EXECUTABLE ARCHITECTURES AND THEIR APPLICATION TO A GEOGRAPHICALLY DISTRIBUTED AIR OPERATIONS CENTER

THESIS

Robert J. Beal, Captain, USAF

Jeremy P. Hendrix, Captain, USAF

Garth P. McMurray, Captain, USAF

William C. Stewart, Civilian, USAF

AFIT/GSE/ENY/05-M03

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GSE/ENY/05-M03

EXECUTABLE ARCHITECTURES AND THEIR APPLICATION TO A GEOGRAPHICALLY DISTRIBUTED AIR OPERATIONS CENTER

THESIS

Presented to the Faculty

Department of Aeronautics and Astronautics

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Systems Engineering

Robert J. Beal, B.S., Mechanical Engineering
Captain, USAF
Jeremy P. Hendrix, B.S., Computer Science
Captain, USAF
Garth P. McMurray, B.S., Computer Engineering
Captain, USAF
William C. Stewart, B.S., Mechanical Engineering
Civilian, USAF

March 2005

AFIT/GSE/ENY/05-M03

EXECUTABLE ARCHITECTURES AND THEIR APPLICATION TO A GEOGRAPHICALLY DISTRIBUTED AIR OPERATIONS CENTER

Robert J. Beal, B.S., Mechanical Engineering

Jeremy P. Hendrix, B.S., Computer Science

Garth P. McMurray, B.S., Computer Engineering

William C. Stewart, B.S., Mechanical Engineering

Approved:

Lt Col John M. Colombi (Chairman)	Date
Dr. David R. Jacques (Member)	Date
Maj Joerg D. Walter (Member)	Date

AFIT/GSE/ENY/05-M03

Abstract

Integrated Architectures and Network Centric Warfare represent two central concepts in DoD's on-going transformation. The true power of integrated architectures is brought to bear when they are combined with simulation to move beyond a static representation and create an executable architecture. This architecture can then be used to experiment with system configurations and parameter values to guide employment decisions. This thesis applies the methodology of Dr. Alexander Levis, former Chief Scientist of the Air Force, to the static architecture representing the Aerospace Operations Center (AOC). Using Colored Petri Nets and other simulation tools, an executable architecture for the AOC's Air Tasking Order (ATO) production thread was developed. These models were then used to compare the performance of a current, forward deployed AOC configuration to three other potential configurations, which utilize a network centric environment to deploy a portion of the AOC and provide reach-back capabilities to the non-deployed units. Performance was measured by the amount of time required to execute the ATO cycle under each configuration. Communication requirements were analyzed for each configuration and stochastic delays were modeled for all transactions in which requirements could not be met due to the physical configuration of the AOC elements. All four configurations were found to exhibit statistically different behavior with regard to ATO cycle time.

Dedication

Our team would like to dedicate this work to our families who supported us throughout our time at AFIT. This work would not have been possible without their sacrifices.

Acknowledgements

We would like to thank Capt. Matthew Ewoldt, who got his hands dirty with $\LaTeX$.

Maj. Paul Lambertson's operational insight on AOC operations was invaluable in constructing our executable architecture products as well as in designing our simulation experiments.

Finally, we would like to thank our thesis advisor, Lt. Col. John Colombi, without whom we would have neither a purpose nor a viewpoint.

Robert J. Beal
Jeremy P. Hendrix
Garth P. McMurray
William C. Stewart

Table of Contents

Page

List of Figures

Figure		Page

Figure		Page

List of Tables

Table Page

EXECUTABLE ARCHITECTURES AND THEIR APPLICATION TO A GEOGRAPHICALLY DISTRIBUTED AIR OPERATIONS CENTER

I. Introduction and Problem Statement

1.1 Introduction

The concept of "transformation" is perhaps the most transcendent idea within today's Department of Defense (DoD). Whether it is in strategic forecasting, acquisition, or contingency operations, transformational concepts abound. Perhaps the grandest vision centers on the concept of Network Centric Warfare (NCW). Enabled by cutting edge technology, the high-bandwidth networks of the Global Information Grid, and visionary command and control (C2) doctrine, NCW aims to move the very basis of warfighting operations away from "platform-centered" thinking and toward a "capability-centered" model. In doing so, proponents of NCW claim, the DoD will generate "increased combat power by networking sensors, decision makers, and shooters to achieve shared awareness, increased speed of command, higher tempo of operations, greater lethality, increased survivability, and a degree of self-synchronization." [13, 2].

NCW is a far-reaching concept with implications for virtually every facet of DoD operations. One of the most obvious potential applications of an information-enabled force is the ability to conduct distributed operations. As the saying goes, "it is cheaper to move electrons than people." This concept pre-dates the formal notion of NCW and has been experimented with extensively both within the DoD and in private enterprise. It has even given rise to the terms "virtual organizations" and "virtual collaboration." A virtual organization exists only as long as necessary to complete a particular task. Once completed, the organization can be disbanded and its resources freed to work on other tasks. While in existence, the virtual organization's members can collaborate in a

virtual workspace utilizing networking to make their physical location far less important. This allows leadership to assign the most applicable assets to the most important tasks, regardless of physical location. [13, 20,39]

The United States Air Force (USAF) is uniquely positioned to take advantage of early advancements in the realization of the NCW concept. The ability of aircraft to operate largely without regard to physical terrain and to move within a theater of operation at a high rate of speed allows them to rapidly exploit information advantages. Given the degree of air superiority, if not air supremacy, which USAF assets often enjoy in today's operations, information about enemy forces is perhaps the most important element to a successful air campaign. Indeed, recent experiences in Afghanistan, where bombers delivered precision-guided ordinance on coordinates provided by Army personnel equipped with laptop computers in time frames measured in tens of minutes, have highlighted the high degree of effectiveness that can be obtained. Effects that once required massive firepower are now achieved through rapid maneuver and precision, each enabled by information. [9, 19]

Enhanced battlespace awareness, the same "shared awareness" which NCW aims to create, is a force multiplier at all levels. Even greater potential lies in moving this revolution up the chain of command from the tactical engagement level to more strategic decision making layers. The concepts of virtual organizations and virtual collaboration lend themselves directly to the USAF's time-tested principle of "centralized control, decentralized execution." A geographically dispersed virtual organization charged with C2 efforts is perhaps the ultimate example of decentralized execution. While maintaining the tenet of centralized control, this concept offers leadership the potential for a more tailored organization which can be put in place more rapidly and at decreased cost.

The idea of an Air Operations Center (AOC) structured in this manner is not new. The first known exercise of the concept came during a UNIFIED ENDEAVOR exercise held in 1995, and it has been similarly tried in BLUE FLAG C2 exercises. [16, 20] Proponents argued that the so-called "Split AOC" would offer a reduced forward footprint,

which would translate to more rapid, lest costly deployment. Further, it was argued that by reducing the forward footprint, fewer personnel would be exposed to force-protection risks and the risk of a single successful attack crippling C2 operations would be reduced.

These efforts met with mixed results, with most criticisms centering on the lack of robust communications and interoperable C2 systems. Indeed there was evidence that the increased requirements for communications equipment and bandwidth, some of which had to be commercially rented, actually made the Split AOC more costly. [16, 25] This is reflective of the stove-piped nature of the forces that existed at the time. These are exactly the issues that NCW aims to overcome. By making networked communications and interoperability characteristics of daily operations and not special requirements to be stood up for a particular occasion, this cost element should be reduced if not eliminated.

Even if these technical obstacles are overcome, the question of the effectiveness of the Split AOC remains. On the one hand, there is evidence from previous exercises that the products of such a configuration are less than optimal. This is not surprising given the difficulties outlined above which these efforts have encountered; however, the elimination of these difficulties would not likely lead to an equal degree of effectiveness. All Joint and Service doctrine concerning the AOC assumes a single, unified command structure. Take away the relatively simple technological limitations and you are still faced with this cultural memory. The difficulties that remained, in the near term at least, would be the inherent cost of the loss of face-to-face contact.

There is also the potential for degradation in effectiveness as a result of a loss of synergy. Discussions with personnel who have operational experience in AOCs highlight the numerous occasions in which a problem has been solved or even avoided because a member of one division happened to overhear another division's conversation, or noticed a discrepancy on a display board in another cell. These chance encounters will not occur if the divisions are geographically separated and communicate only via intentional virtual collaboration. This introduces a further requirement for the operational employment of a Split AOC, the need for documented, mature processes which ensure that these "lucky

breaks" are not necessary. On the other hand, the potential also exists for the products of a Split AOC to be not only comparable but even superior to those of a collocated, deployed entity. If the data analysis assets that exist organically within the Continental United States can be leveraged effectively to analyze and synthesize intelligence coming from multiple sources throughout the theater, they may well be able to achieve superior results compared to deployed analysis assets. Of course, a necessary first condition for this to take place is an architecture which overcomes the aforementioned technical limitations.

Thus, the Split AOC concept faces two fundamental issues: First, can an architecture be identified which overcomes the loss in product quality that can be expected without face-to-face collaboration? And second, can such an architecture provide operational benefits of sufficient magnitude to justify such a fundamental shift in how the USAF accomplishes expeditionary C2?

1.2 Problem Formulation

1.2.1 General Problem. Beginning with the premise that geographically separating an AOC will impact both the timeliness and quality of the products developed, a cost-benefit type analysis is required in order to fully address the merit of the idea. In order to truly perform such an analysis for this shift in operations it is necessary that both areas of impact be quantified through some objective measure. Arriving at such measures requires three fundamental activities be performed:

1. Develop an architectural representation for the structure of both the collocated and geographically distributed AOC functionality.
2. Generate executable models from this architecture for use in simulation analysis.
3. Define Measures of Effectiveness to be extracted from the data generated by the execution of these models.

1.2.2 Scope and Assumptions. An operational AOC is an extremely large and complex organization, and to attempt to develop a single, monolithic model of its

operations would be beyond the scope of this thesis. Accordingly, for the purposes of this thesis, our scope will be reduced to the Air Tasking Order (ATO) development thread. This is a major, although by no means the only, function of the AOC and will be described in greater detail in Chapter 2.

Additionally, it is necessary to address some of the organizational issues of the AOC at a higher degree of abstraction. As will be described later, the ATO production thread is carried out by five divisions, each of which is comprised of a varying number of teams. For the purposes of this thesis, the assumption is made that divisions act as single organizations, i.e. all teams within a division are collocated. It is recognized that this is an artificial limitation, as there may well be an operational advantage to distributing not only the divisions but the teams within the divisions as well. However, this creates an extremely large number of potential configurations, and so for the purposes of this thesis we will focus on modeling at the division level. Further research will likely focus on alternate configurations for a Split AOC.

1.2.3 Thesis Goal. As mentioned earlier, a cost-benefit analysis is a necessary first step in evaluating the merit of the Split AOC concept. The benefit side of this analysis will, in all likelihood, ultimately be monetized so that it can be expressed in terms of dollars available for application elsewhere. It is not the intent of this thesis to arrive at a measure for the benefit. Such an effort would be heavily dependent on the physical architecture ultimately employed, as well as the complexities of contract vehicles and opportunity costs. We will aim instead to quantify, to the greatest degree possible, the time and quality impacts of employing the Split AOC versus the traditional collocated AOC.

II. Background

The foundation of this research is the interweaving of four seemingly unrelated subject areas. These are the Air Operation Center (AOC), Communication Theory, Department of Defense Architecture Framework (DoDAF), and Executable Architectures. The AOC is the system of interest. Specifically, how does a geographically distributed AOC impact effectiveness? In order to be able to make any assessments, Measures of Effectiveness (MOEs) must be selected. The MOEs that have been selected stem from research done in communication theory. DoDAF provides the capability to create a universal and unambiguous description of the system. The DoDAF products are then used to create an executable architecture which enables analysis of both the originating DoDAF products as well as the system of interest. The necessary background information of these four subject areas is provided within this chapter.

2.1 Air Operations Center

In order to be able to correctly architect and model an AOC, an in-depth discussion of its operational procedures and organizational structure is required. According to the Air Force Operational Tactics, Techniques, and Procedures (AFOTTP) manual for Air and Space Operations Center, ”The AOC is the operational-level command and control (C2) center that provides the Commander of Air Force Forces with the capability to direct and supervise the activities of assigned and attached forces and to monitor the actions of both enemy and friendly forces” [1, 1-1]. This research effort will be concentrated on the air tasking cycle which utilizes all available intelligence and guidance to produce the Air Tasking Order (ATO) for all available air assets. The ATO development is a key thread of overall AOC operations with direct implications of utilizing reachback capability for a geographically distributed AOC.

2.1.1 AOC Operational Procedures. The air tasking cycle and battle rhythm are very closely related to AOC operational procedures. At a very high level of abstraction,

the air tasking cycle is the process which utilizes Joint Force Commander (JFC) and Joint Force Air Component Commander (JFACC) guidance in order to develop targets, allocates assets, and finally produces an Air Tasking Order (ATO). To complete the cycle the ATO must be executed and assessed. The assessment of the ATO will feed back into the cycle by influencing the objectives, desired effects, and JFC and JFACC guidance. This cycle is used to establish the battle rhythm. The battle rhythm is the timing and synchronization that needs to occur throughout the air tasking cycle.

2.1.1.1 Air Tasking Cycle. The air tasking cycle is a six step process as shown in Figure 2.1. The first four steps of the process deal with developing the Air Tasking Order and will be the focus of our study. The significant products of each step are shown between the steps. The Air Tasking Cycle is accomplished through collaboration between divisions, indicated by the organizational resources at the center of the figure.

The first step is to produce objectives, effects, and guidance. This planning phase begins with the JFC sending a formal written letter to the JFACC. This letter describes the next ATO's priorities and apportionment for airpower missions. The strategy division, with collaboration of Liaison Officers from the JFC staff, submits recommended guidance and apportionment. This input should be in accordance with the JFC guidance letter, making the production of the Air Operations Directive (AOD) more rapid and efficient. The strategy division takes the JFC guidance letter and JFACC interpretation of the guidance letter and writes the AOD. The JFACC interpretation is based on the Intelligence, Surveillance, and Reconnaissance Division's (ISRD) continuously updated intelligence preparation of the battlespace. All of this guidance is taken into account in the preparation of the Joint Air Operations Plan (JAOP).

Target Development begins with the strategy division prioritizing operational tasks to meet the JFACC written guidance. After the operational tasks have been prioritized, component representative will nominate target sets mapping back to the published tasks. The ISRD's targets and combat assessment team merges all of the nominated targets

Figure 2.1 Air Tasking Cycle [1, 1-12]

into a proposed target list for the Target Effects Team (TET). The TET aligns targets with the objectives in a draft Joint Integrated Prioritized Target List (JIPTL). The JFACC reviews and approves the JIPTL during the briefing conducted by the TET working group. Likewise, the JFACC approves the prioritized, facility-level Candidate Target List/Target Nomination List (CTL/TNL) after being adjusted/altered and approved by the JFC.

The third step is to develop weaponeering and allocation solutions. This process begins by the JIPTL returning to the targets and combat assessment team to have each apportioned target weaponeered. Next, the JIPTL goes to the Master Air Attack Plan

(MAAP) process where JFACC resources are matched to each target. The MAAP process can change the cut line based on support considerations.

The next step is to produce and disseminate the ATO. The ATO is developed by combining collection planning and target planning. This ensures the targets selected for the ATO are matched with collection requirements for pre-strike verification and post-strike Battle Damage Assessment. After the ATO, Special Instructions (SPINs) and ISR synchronization matrix are developed, the data is compiled into Theater Battle Management Core System (TBMCS) and electronically transmitted to all users.

Execution planning and Force execution and the assessment process complete the cycle. During the execution phase the JFACC controls operations from the AOC. As the ATO is being executed and Mission Reports are distributed, operational assessments are determined. The air tasking cycle completes one evolution as the Operational Assessment Team meets to determine the effectiveness of air operation and, with JFACC approval, sends recommended air actions for the JFC to consider in his next guidance letter.

2.1.1.2 Battle Rhythm. After entering the execution phase, the AOC will fall into a battle rhythm. The battle rhythm will vary greatly depending on a number of factors and can be different for each exercise and contingency. A notional ATO battle rhythm has been developed in Joint Publication 3-30, Command and Control for Joint Air Operations. The time line for this notional battle rhythm is shown in Figure 2.2.

At any given time there will be four simultaneous ATOs being planned and/or executed. Figure 2.3 shows a typical battle rhythm for ATO development through assessment. The ATO progression is development (ATOs D and E), tasking (ATO C), execution (ATO B), and assessment (ATO A). In the diagram, ATO E begins after ATO A has been executed and is being assessed. Note that these figures are notional and that specific battle rhythms and planning cycles may be different for each component based upon their operational procedures. For example, the Joint Force Special Operations

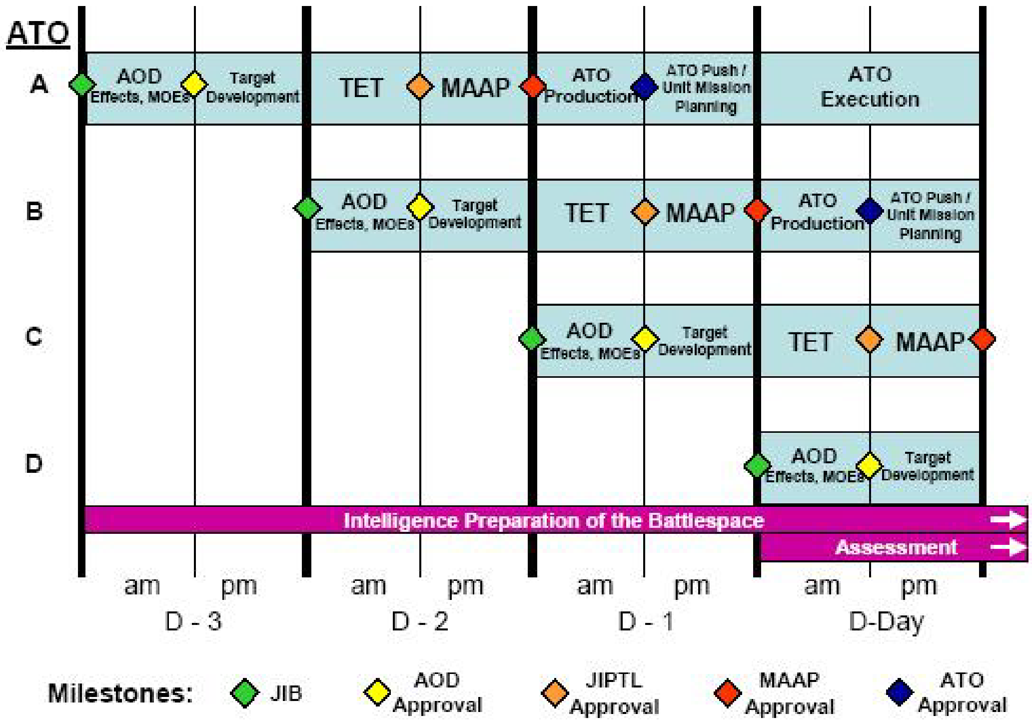


Figure 2.2 Battle Rhythm Time Line [1, 1-13]

Component Commander uses a bottom up planning method that may cause targeting well inside of the AOC planning cycle previously described.

The Air Tasking Cycle and the AOC Battle Rhythm have been presented as linear processes. While this is true at a high level of abstraction, it should be noted that there are many feedback loops throughout the process which are not captured at this level of abstraction. This feedback may be in the form of incomplete or substandard products which go back to the originating team/division for revisions. In discussions with Major Paul Lambertson, who has invaluable experience as Chief of C-17 Tactics in the CAOC, it has been indicated that it is common to go through many revisions of the ATO during the execution day. These abnormalities of the air tasking cycle have been noted and are well suited for future studies. The Air Tasking Cycle is assumed to be a perfectly linear process for our study.

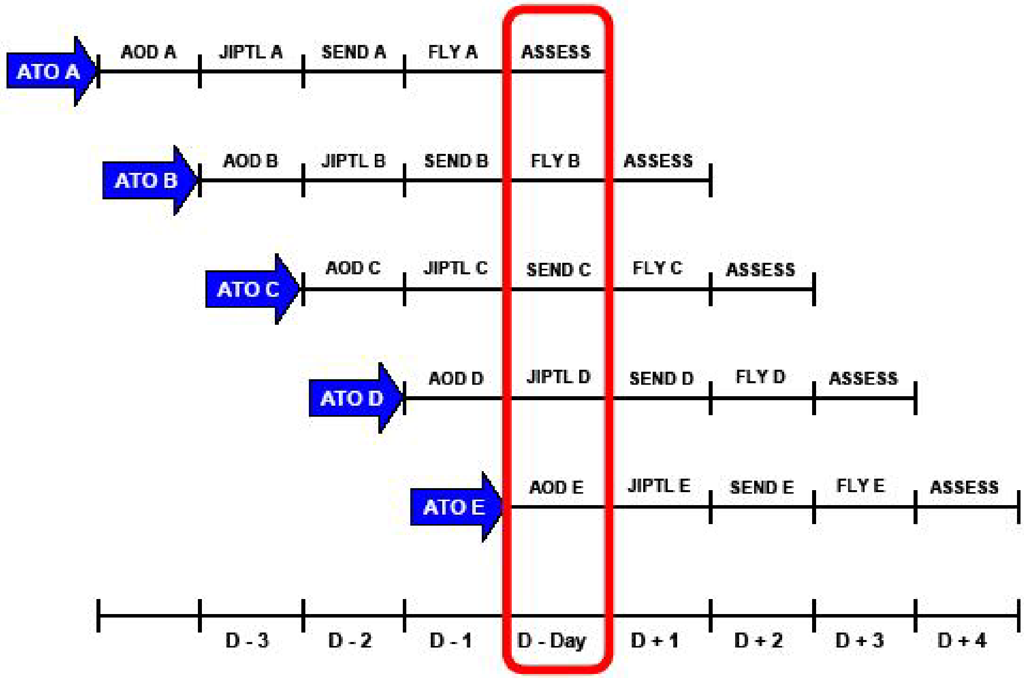


Figure 2.3 ATO Battle Rhythm [1, 1-14]

2.1.2 AOC Organizational Structure. The AOC is composed of divisions responsible for strategy, plans, operations, intelligence, and air mobility. Each of these five divisions is composed of teams and/or cells. Various sources emphasize different aspects of the AOC, but the AFOTTP manual for Air and Space Operations Center provides the most comprehensive description of the AOC and is an excellent source for additional information.

Each division is supported by specialists in communications, weather, Information Operations, and other functional areas as needed. This basic structure of an AOC is shown in Figure 2.4. The size of the AOC can vary greatly depending on the missions being performed and the number of forces involved. The capability of the AOC can also vary from some limited capability in order to perform unopposed humanitarian assistance operations to a fully operational AOC capable of supporting an effort such as Operation Desert Storm or Operation Iraqi Freedom. Regardless of the size or capability level of an

AOC, the need for the JFACC to have a single command and control system to exercise control over forces remains the same.

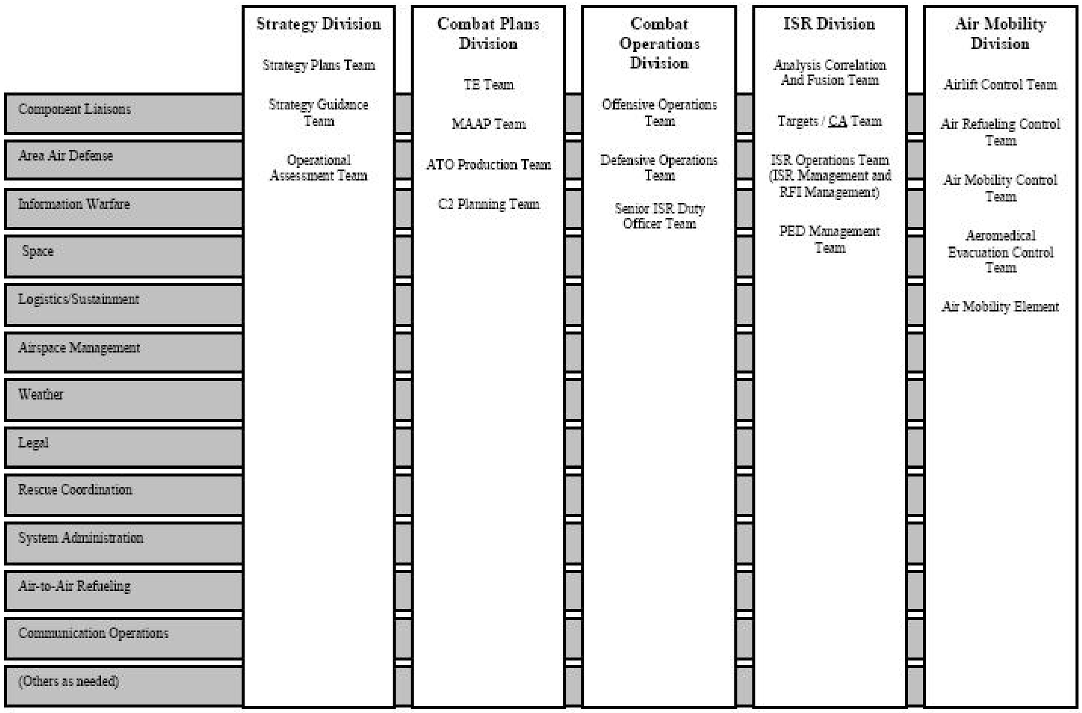


Figure 2.4 Basic AOC Structure [1, 1-4]

2.1.2.1 Strategy Division. The Strategy Division is responsible for the "long-range planning of air, space, and information operations to achieve theater objectives by developing refining, disseminating, and assessing the JFACC air and space strategy" [1, 3-1]. Typically the strategy division is organized into three functional teams: Strategy Plans Team, Strategy Guidance Team, and the Operational Assessment Team. The strategy division also has Intelligence, Surveillance, and Reconnaissance Division (ISRD) personnel nested in order to provide direct intelligence support. Figure 2.5 shows the typical strategy division organizational and operational roles.

The strategy plans team main responsibility is to develop and maintain an operational-level, long range joint air strategy that supports the JFC's objectives. The strategy plans team leads the JAOC in the development of the Joint Air Operations Plan

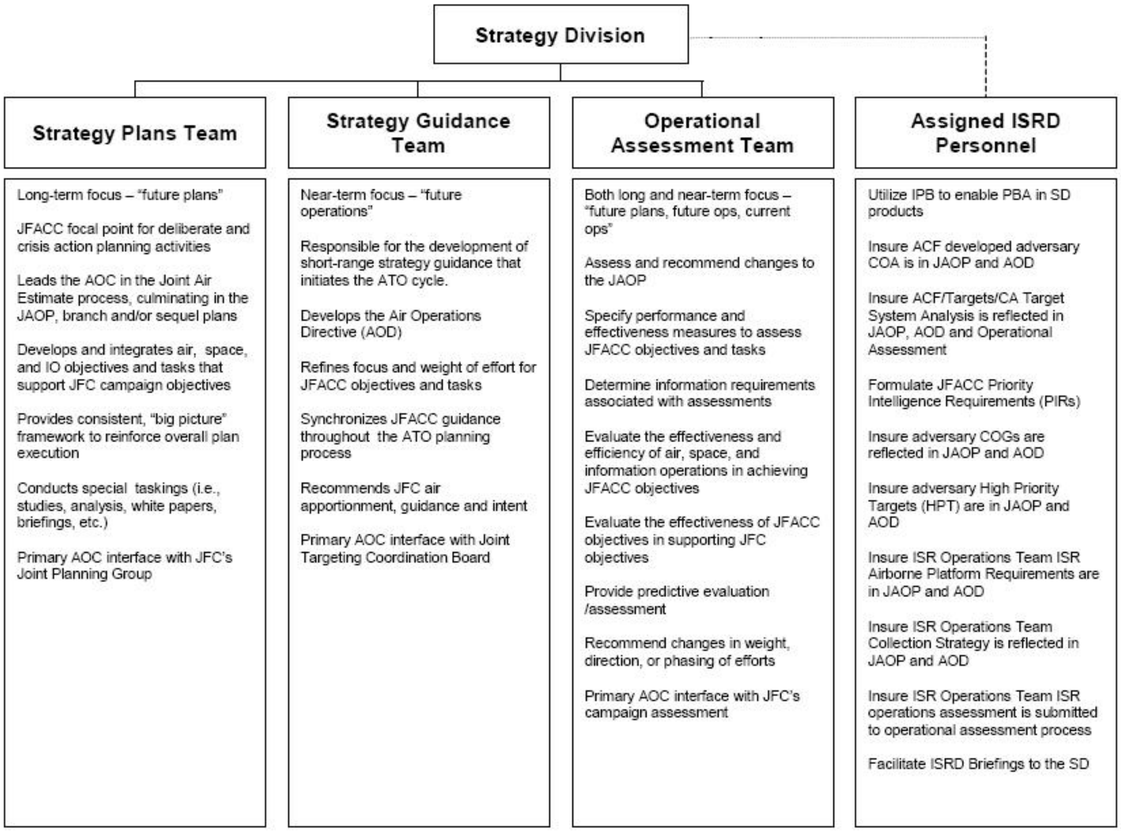


Figure 2.5 Strategy Division Team Construct [1, 3-4]

(JAOP) which includes a prioritized, effects-based targeting scheme. The strategy plans team also is responsible for the development of operational-level guidance in the Air Operations Directive (AOD). The strategy plans team can also act as the JFACC's action group for Course of Action (COA) development and strategy related special projects.

The strategy guidance team is responsible for the AOC's transition from operational-level planning to tactical-level planning. The strategy guidance team also assists in the detailing of the daily guidance in the AOD. This team provides short range guidance from 48 to 72 hours prior to ATO execution. This guidance is contained in the AOD.

The last team under the strategy division is the operational assessment team. The operational assessment team is involved in all aspects of strategy development while concentrating on evaluating the effectiveness and efficiency of air, space and information

operations. This team assists the other strategy division teams in producing the JAOP and AOD.

2.1.2.2 Combat Plans Division. The Combat Plans Division (CPD) develops plans for near-term air and space operations. Typically, this is the 48 hour time period prior to an ATO execution. The combat plans division uses JFC and JFACC-approved guidance received through the strategy division to develop detailed plans for air and space operations. The near-term planning is accomplished through the four functionally oriented teams that compose the combat plans division. The combat plans division's organizational structure is shown in Figure 2.6.

Figure 2.6 Combat Plans Division [1, 4-1]

"The Targeting Effects Team (TET) is the linkage between the JFACC's vision and its practical application" [1, 4-5]. The TET's primary mission is to ensure the daily target selection process reflects the guidance found in the AOD. This includes the production of the daily draft Joint Integrated Prioritized Target List (JIPTL) and inputs into the JFACC's Component Prioritized Collection List (CPCL). A process called the Strategy-to-Task Methodology is used to guarantee that each target on the JFC's JIPTL can be traced directly back to a JFC campaign objective. The effects on JIPTL targets can be kinetic or non-kinetic. For example, kinetic effects would be used for either a hard or soft kill while non-kinetic effects might include a leaflet drop on a ground unit. The JIPTL must include all of the joint force's prioritized targets, even if it is a non-kinetic effects target. The JIPTL should also clearly indicate the desired effect of *all* prioritized targets. In addition, the JIPTL may include both fixed and mobile targets.

The TET also provides inputs to the Strategy Plans/Guidance Team for use in the initial development of the AOD. The Target Development Cell in the Intelligence Surveillance, Reconnaissance Division (ISRD) will produce the air and space component Target Nomination List (TNL), and merge this TNL with all other component TNLs into one integrated TNL. The TET uses the integrated TNL and the approved AOD to develop the draft JIPTL. The TET inputs to the JFACC's Component Prioritized Collection List (CPCL) are sent to the ISR Operations Team for collection plan development.

The MAAP Team develops the daily Master Air Attack Plan and loads it into the Theater Battle Management Core System (TBMCS) to use during ATO production. "The Master Air Attack Plan is the JFACC's time-phased air and space scheme of maneuver for a given ATO period and it synthesizes JFACC guidance, desired effects, supported component's scheme of maneuver, available resources, friendly capabilities, and enemy capabilities"[1, 4-21]. The MAAP team is composed of highly trained representatives from across the spectrum of air and space disciplines. In order to accomplish its mission, the MAAP team must maintain clear, two-way lines of communication in order to coordinate with other CPD teams, JFACC staff, Component/Service representatives, and host/coalition representatives.

The ATO Production Team assembles, disseminates, and publishes the daily ATO. The ATO is the primary document containing the guidance, tasking, and apportionment of all available air and space resources. The AFOTTP 2-3.2 outlines many responsibilities for the ATO Production Team, some of which have been listed below:

- Review the most current version of the Rules Of Engagement, all detailed execution plans, and supporting SPINs required to develop and produce the ATO.
- Create and maintain accurate air and space planning databases in the theater battle management system and/or applications. This will normally include regular and periodic data backups. Effect quality control procedures to ensure accuracy of data inputs, worksheets, and other baseline planning materials.

- Input complete, accurate, properly formatted, and timely mission tasking to theater battle management applications using standard ATO formats, as required.
- Accomplish a comprehensive ATO review.
- After approval for release, disseminate the ATO to tasked units and agencies by the most expeditious means available.

The C2 Planning Team is composed of multiple functional teams that correspond to different roles that the JFACC assumes (i.e., Airspace Control Authority, Area Air Defense Commander, Space Control Authority, etc). The C2 Planning team is composed of Airspace Management, Air Defense, C2 Architecture, and C2 Communications Planning Teams. The Airspace Management Planning Team supports both the Combat Operations and Combat Plans Divisions. The CPD Airspace Management Planning Team is responsible for developing the Airspace Control Order (ACO). The C2 Planning Team is also responsible for developing and distributing several other critical documents. Air traffic controllers, air defense personnel, and C2 subject matter experts from service components and coalition partners must be incorporated in C2 Planning Team. The Combat Operations Division (COD) Airspace Management Team is responsible for executing the ACO by making real time changes and deconflicting immediate requests for airspace.

2.1.2.3 Combat Operations Division. The COD adjusts the ATO, as necessary, to respond to battlefield dynamics. The COD publishes changes to the ATO and ACO as necessary. The COD is composed of offensive and defensive operations teams, and the ISR Senior Intelligence Duty Officer (SIDO) Team. The SIDO Team includes assigned ISR support, airspace management, weather, and the air and space component's Rescue Coordination Center. The COD is also the focal point for monitoring the execution of joint and combined operations, such as Joint Air Attack Team, Theater Missile Defense, and Joint Suppression of Enemy Air Defense supported by theater forces. Personnel assigned or attached to the COD support the offensive effort, the defensive effort, or both.

The ATO is typically released 12 hours prior to execution. The COD assumes responsibility of the ATO as soon as it is released.

"The Offensive Operations Team is responsible for executing the ATO and makes command and control decisions to ensure the theater offensive air campaign is executed in accordance with commander's guidance, and in reaction to the current battlespace situation for all offensive missions"[1, 5-6]. The Offensive Operations Team is organized by mission (Close Air Support, Interdiction, Electronic Warfare/Suppression of Enemy Air Defenses, etc.) and is supplemented by weapons systems experts. The Offensive Operations Team continually monitors the battlespace and recommends changes to the ATO based on unforeseen challenges and opportunities.

"The Defensive Operations Team provides C2 battle management within the theater, and oversight of the overall execution of theater air and missile defense operations"[1, 5-47]. Defensive Operations Team personnel monitor the status of air defense assets and assist the Senior Offensive Duty Officer to provide offensive targeting and direction to attack assets. "The Defensive Operations Team normally manages the data link network, provides the JFACC with a consolidated and accurate air/battlespace picture, and provides direction to attached units relative to Air Defense Operations, and changes to Air Defense Warning status"[1, 5-47]. In order to achieve their mission, defensive operations personnel have access to a wide variety of communications equipment used to manage command and control assets for the entire air defense effort.

ISR processes in the COD will be led by the Senior Intelligence Duty Officer (SIDO). The SIDO's Team provides Situation Analysis, Target Duty Officers, and ISR Operations specialists. "The SIDO Team is responsible for real-time situational and predictive analysis of the adversary in the battlespace, monitoring and dynamically adjusting ISR collection plans, monitoring current day's ATO targets and recommending reroles and tracking and highlighting dynamic targets and time sensitive targets"[1, 5-71]. The SIDO Team is heavily dependent on regular updates from the ISRD.

2.1.2.4 ISR Division. The Intelligence, Surveillance, Reconnaissance Division (ISRD) is responsible with providing the JFC and JFACC situational awareness of the adversary in order to maintain an accurate and up-to-date target list. The ISRD mission includes monitoring "current and emerging enemy capabilities, threats, courses of action (COA) and centers of gravity with predictive and actionable intelligence, and to provide the JFACC with ISR operations management and targeting intelligence support"[1, 6-1]. As can already be seen by the discussion of previous divisions, the ISRD has nested portions in all of the other four divisions. The ISRD provides critical information to the other four divisions as they plan and execute air and space operations. This information not only enables the commanders objectives to be accomplished, but also provides an assessment of previous operations. The responsibilities of ISR personnel nested inside the other divisions is summarized in the list below:

- Provide analysis of the enemy and a common threat picture to the JFACC and Staff Planners, other JAOC divisions and other AF elements in theater.
- Provide, in conjunction with the Strategy Division, Combat Plans Division and Combat Operations Division, for planning and executing airborne ISR operations and providing combat ISR support to planning, execution, and assessment activities.
- Direct the JAOC's distributed and reachback ISR processes by working with national agencies, Air Force organizations, and the processing, exploitation, and dissemination architecture to optimize ISR contributions to the effort.
- Provide direct targeting support to the Air Tasking Cycle in response to JFACC guidance.
- Provide all-source intelligence support to other JAOC divisions to enhance the execution of their core processes.
- Ensure actionable, decision quality, all-source Intelligence Preparation of the Battlespace (IPB) and threat information depicted in the JFACC and JAOC picture

of the battlespace is consistent with that used by National, Joint, Component, and Theater entities. Aggressively act to resolve significant differences.

- Ensure Air Force ISR is optimally managed to operate within the context of a large and complex national and joint intelligence architecture. Serve as the focal point for multiple nodes to work together in order to meet the high demand for information.

"The Analysis, Correlation and Fusion Team (ACF) conducts dynamic intelligence preparation of the battlespace (IPB) that provides the context for understanding the adversary's intentions and supports the application of Predictive Battlespace Awareness (PBA)" [1, 6-5]. The ACF can be organized in different ways depending on the situation and current needs. A functionally organized ACF is well suited for non-traditional or asymmetric treats. It is critical for the ACF to maintain a close working relationship with other branch intelligence elements.

The Targets/Combat Assessment Team coordinates targets and combat assessment functions for the JFACC. The team is comprised of two primary cells, the Target Development Cell and the Combat Assessment Cell. There are targeteers assigned throughout the other JAOC divisions to ensure continuity in the targeting effort. Each step of the ATO cycle contains elements of the targeting process. "Targeting as a whole is an integration of strategy, plans, intelligence, and operations and can be applied at the strategic, operational, or tactical level"[1, 6-45]. The targeting process closely aligns itself with the air tasking cycle phases. Even though the air tasking phases are shown sequentially, it is not uncommon for targeteers to perform several of the phases simultaneously.

ISR operations are the process of developing ISR strategy and plans and executing and adjusting those plans to satisfy theater intelligence requirements. The ISR operations process in the JAOC is the responsibility of the ISR Operations Team and is accomplished through a collaborative effort of collection managers, reconnaissance and surveillance planners, Chief Processing, Exploitation, and Dissemination (PED), and PED centers to ensure ISR operations are synchronized with joint operations.

"The PED Management Team is the ISRD focal point for implementing, coordinating, and maintaining PED support from units/agencies outside the JAOC"[1, 6-147]. Responsibilities include: directing, managing, and coordinating all PED activities in support of the JAOC. The PED Management Team monitors ISR assets and PED mission execution, identifies discrepancies in the PED mission, and institutes control measures to correct or improve the PED process. The process of gathering and metrics for ISR Operations assessment is manpower intensive due to the lack of automation.

2.1.2.5 Air Mobility Division. The Air Mobility Division (AMD) is comprised of five teams: Airlift Control Team, Air Mobility Control Team, Air Refueling Control Team, Aeromedical Evacuation Control Team, and Air Mobility Element. The Air Mobility Element deploys as a liaison element of HQ Air Mobility Command Tanker/Airlift Control Center (TACC) in controlling Air Mobility Command (AMC) missions. Figure 2.7 depicts the standard AMD organization. The Director of Mobility Forces is responsible for integrating the total air mobility effort for the JFACC, and providing direction to the AMD to execute the air mobility mission.

The AMD will plan, coordinate, task, and execute the intra-theater air mobility and air refueling mission. The AMD provides for integration and support of all air mobility missions. This integration and support of all air mobility missions incorporates the AMD in all phases of the air tasking cycle. The AMD coordinates movement requirements with the JFC movement requirements and control authority, the theater Air Mobility Operations Control Center, and the AMC TACC. As directed by the Director of Mobility Forces, the AMD will task assigned/attached theater air mobility forces.

The Airlift Control Team (ALCT) is the source of intra-theater expertise within the AMD. The ALCT brings theater airlift functional expertise from the theater organizations to plan, schedule, coordinate, and manage theater airlift operations and airspace. The ALCT develops and integrates the airlift schedule into the ATO and ACO. The ALCT processes validated airlift requirements from the Theater Movement Validation

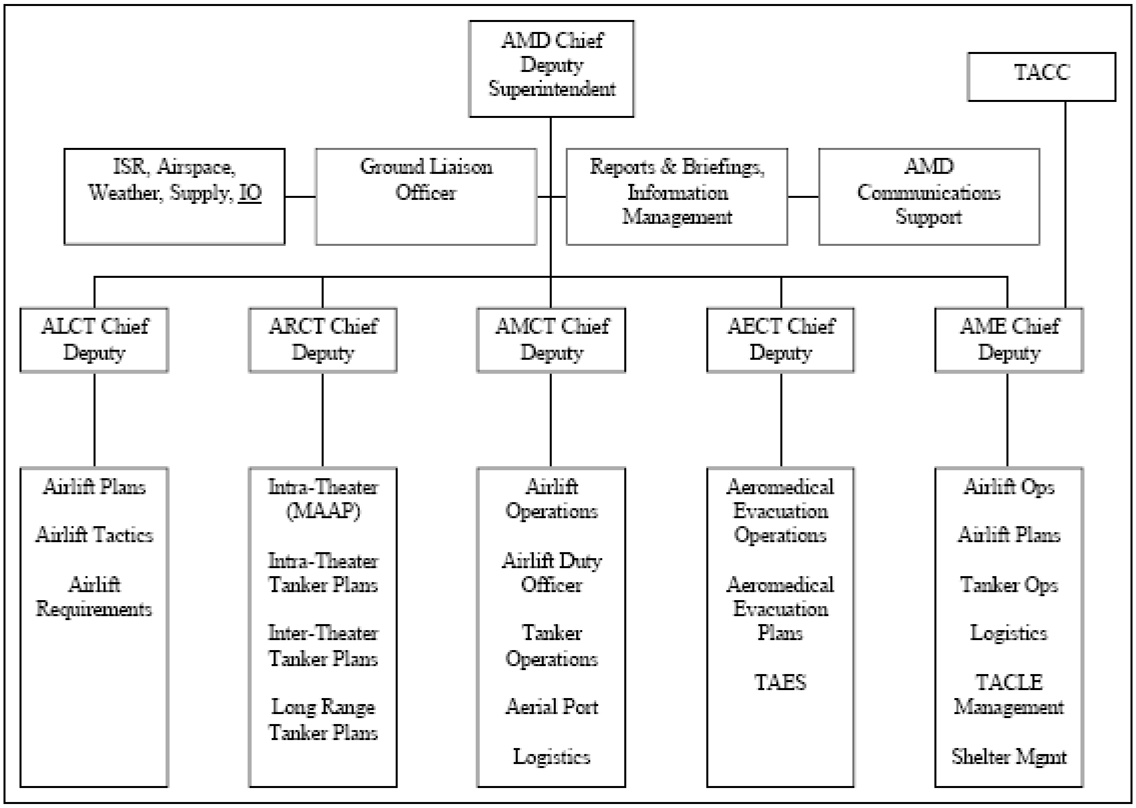


Figure 2.7 Air Mobility Division Organization [1, 7-2]

Authority or Joint Movement Center and develop the intra-theater airlift schedule. The ALCT integrates the intra-theater and inter-theater airlift schedules and airspace usage into TBMCS and validates the integration of the airlift schedule into the ATO/ACO. The ALCT will integrate its activities with all other AMD teams and support functions as much as possible to support the total mobility effort.

The Air Refueling Control Team (ARCT) plans, tasks, and schedules air-refueling missions under the control of the JFACC to support theater air and space operations. It coordinates the air refueling planning, tasking, and scheduling to support air bridge and global attack missions. The ARCT will integrate its activities with the CPD to support the master air attack planning (MAAP) and ATO/ACO production development. The ARCT incorporates the capability to accomplish intra-theater and long-range air-refueling planning and coordinate inter-theater air refueling planning.

The Air Mobility Control Team (AMCT) serves as the centralized source of theater air mobility command, control, and communications (C3) during mission execution. The AMCT will direct, or redirect as required, air mobility forces in concert with other air and space forces to respond to requirement changes, higher priorities, or immediate execution limitations. The AMCT will deconflict all air mobility mission and airspace usage operations into, out of, and within the area of responsibility. The AMCT will maintain execution process and communications connectivity for tasking, coordination, and mission forces.

The Aeromedical Evacuation Coordination Team (AECT) is responsible for operational planning, scheduling, and execution of scheduled and unscheduled Aeromedical Evacuation (AE) missions, and positioning of AE ground support assets. The AECT maintains both secure and non-secure communications links with all AE components. The AECT, in conjunction with the ALCT, assigns resources required to execute the AE mission and ensure integration into the ATO/ACO. The AECT coordinates closely with the Rescue Coordination Center to establish a proactive stance for AE missions following Combat Search and Rescue recoveries. The AECT develops plans and strategies and determines number and location of AE assets required to support operations.

2.1.3 Organizational Involvement in the Air Tasking Cycle. Each of the five divisions is an integral part of the air tasking cycle. Often their areas of responsibilities overlap, creating a need for communication and coherency between the divisions. Figure 2.8 shows which steps of the air tasking cycle each division is involved. The ISR division as well as the air mobility division is involved throughout the air tasking cycle. Every step of the six step process requires the involvement of at least three divisions with the exception of target development, which requires four of the five divisions. It is assumed that even though each of the divisions is composed of multiple teams, the divisions must remain collocated. The relationship between the organization structure of the AOC, the air tasking cycle, and the battle rhythm provide the foundation to be able to build a static and executable architecture for the AOC.

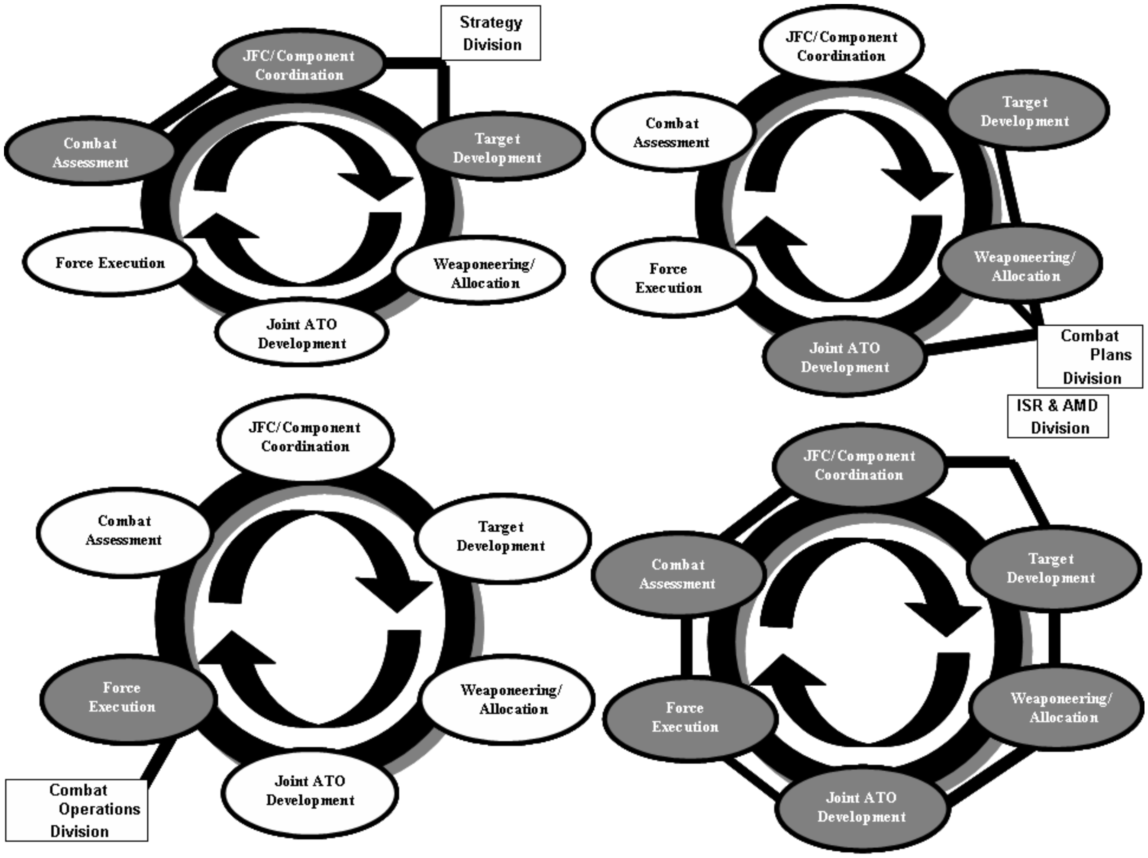


Figure 2.8 Division Involvement in the Air Tasking Cycle

2.2 Communication Theory

2.2.1 Introduction. What is lost in the transfer of information when face-to-face communication is not possible? It is important to understand how the communication medium can influence group processes and outcomes. Not only how the communication medium affects the timeliness of the information, but how the quality of outcomes is affected. Research conducted in parallel with our study was conducted by a group of Intermediate Developmental Education (IDE) students: Majors Brian Baude, Shannon Beggeman, Christopher Eichorst, Christopher Lindell, and Thomas "Lou" Rauls. Their thesis titled "Net-Centric Analysis of Air Operations" provides a significant amount of background to our research on how communication effects the ATO Cycle in a Split-AOC configuration. The following sections elaborate on applicable

research conducted on numerous disciplines including and not limited too, communication, decision science, organizational behavior, business practices, computer science, and most notably psychology.

2.2.2 AOC Communication Requirements. As mentioned in section 2.1, there are many teams/individuals contributing to the ATO cycle. To effectively model distributed functions in the ATO cycle, the communication dynamics must be documented. A collocated AOC has many levels of potential communication from face-to-face to email and when one fails, another can be easily substituted. In a distributed environment, face-to-face communication is replaced by additional virtual tools. The loss of fidelity due to distributed communication must be quantified to effectively represent the distributed AOC. To do this, information exchange requirements (IER) are identified for every communication. These requirements encompass both the information being exchanged and the types of supporting communication media. A prioritized list of media can be created for each communication which allows some flexibility to the distribution of functionality and allows us to apply meaningful values to the resulting degradation as lower priority options are employed. [16, 30]

2.2.3 Effects of Communication Media. To bring context to our problem of identifying IERs of the AOC, let us first examine communication in general. Over the last 40 years much has been written on communication, but much of it is not straightforward. [18, xiii] However, much of the research which quantifies communication needs has centered on data transfer, which is useful to determine network-sizing requirements, but does little to evaluate the effect of communication media on human communication needs such as group processes and outcomes. [18, xiii] [16, 32]

2.2.3.1 Virtual Collaboration. Examining the ATO cycle of the AOC's current configuration, one quickly realizes the reason for a collocated AOC is to allow members of many organizations to come together (such as in face-to-face briefings) to

produce products (e.g. MAAP, JIPTL, ATO) which eventually lead to outcomes (e.g. successful missions, effects based operations, bombs on targets). The communication theory that centers on group processes and outcomes is referred to as "collaboration." To seriously study the prospects of split-configurations of an AOC, one must understand how the type of communication media can affect group collaborations. According to Wainfan and Davis, "Virtual Collaborations are collaborations in which the people working together are independent in their task, share responsibility for outcomes, are geographically dispersed, and rely on mediated, rather than face-to-face communication to produce an outcome, such as a shared understanding, evaluation, strategy recommendation, decision, action plan, or other products." [18, xi]

Video Tele-Conferencing (VTC), Audio-Conferencing (AC) (i.e. telephone), and Computer-Mediated Communication (CMC) are different media forms of virtual collaborations. Much of the research conducted in virtual collaboration focus on comparing each media to face-to-face (FTF) communication.

2.2.3.2 Problems and Benefits of Virtual Collaborations. Many leaders will note they prefer FTF communication because they place high value in "looking them in the eye." Military Commanders often note the importance of reading someone's reactions. Often information is sketchy, and its quality can only be communicated through face-to-face communication where the commander can see the speaker's confidence or lack thereof. [16, 28] Nevertheless, there are benefits to virtual collaboration which include:

- Broadening Reach: Meetings can include participants that are geographically dispersed.
- Adaptability: People can be added quickly to a meeting when their participation becomes apparent.
- Time and Money: It is easier to move electrons than people. Moving people requires significant logistics and support both during the move and at the deployed location.

- Safety: Fewer people need to be in the theater of operations, a dangerous location.
- Survivability/vulnerability: More than a single (un)lucky missile is required to destroy the AOC.

2.2.3.3 Relating Media Types. The following sections explain the differences between communication media. Table 2.1 characterizes the different types of media in simple terms and Figure 2.9 shows notational relationships between types of communication media.

Table 2.1 Characterization of Media [17, 4]

Mode	**Defining Characteristics**	**Examples**
Videoconference (VC)	Useful real-time images and voices of other participants; may include other shared images/text.	Group videoconferencing in dedicated rooms; desktop videoconferencing.
Audio Conference (AC)	Voice communication, but no useful real-time video images of other participants; may include other shared images, data, and text.	Phone calls, conference calls, or conference calls where people are also sharing views of images or documents.
Computer-mediated communication (CMC)	Text, images and other data received via computer, without effective real-time voice or video images of other participants.	E-mail, chat rooms, discussion boards, text messaging, instant messaging, shated databases, application-specific groupware.

Video Tele-conferencing vs. Face-to-face communication. VTC can be thought of as communicating with interactive video. Participants may also use graphics and/or whiteboards to display information. There are two types of VTC: “classic VTC” when groups gathered in dedicated conference rooms, and desktop VTC which will be discussed later.

When comparing classic VTC to FTF communication one must first note that VTC image quality has improved since the early studies. However, today it still can be difficult

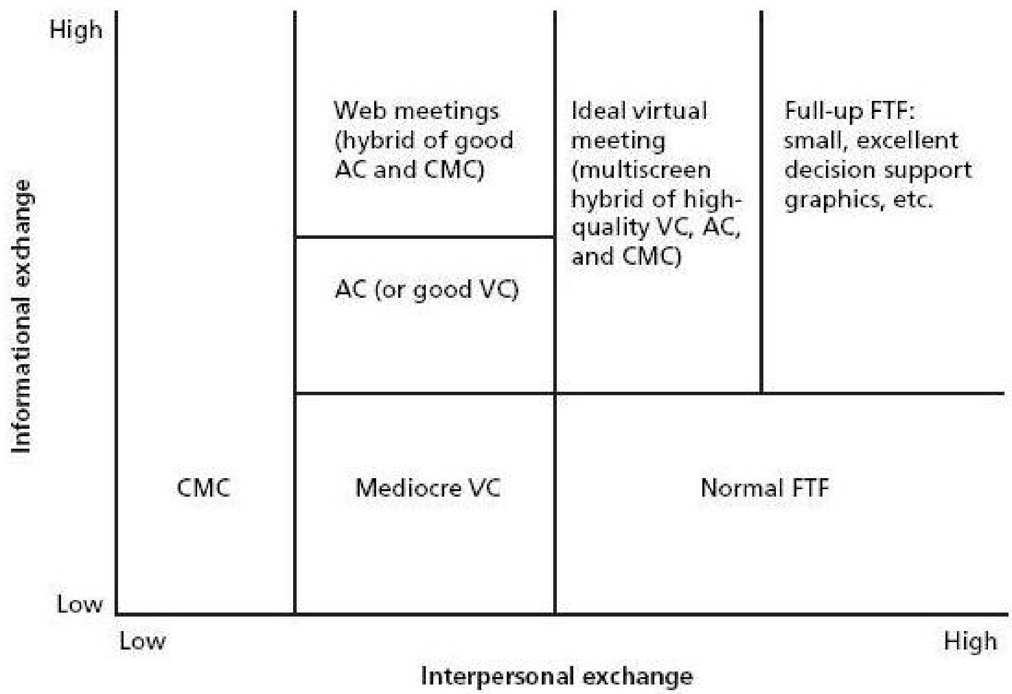


Figure 2.9 Notational Relationships [17, 66]

to maintain eye contact and it is challenging to interpret body language and gestures, especially as the number of participants increases. Wainfan and Davis also suggest, “mediated communication limits nonverbal, paraverbal, and status cue and reduces the ‘richness’ of the information exchange.” [18, 19]

Other research suggests that VTC tends to be less social and more task oriented [15, 63-74] which is good for producing a product. However, more advance preparation is necessary, and meetings conducted via VTC take longer than FTF. [24] The most likely cause of VTCs taking longer is participants’ difficulty in conveying information and the structured communication flow often associated with VTCs.

As for desktop VTC, software called Group Support Software is used to view and operate common screens (e.g., PowerPoint Presentation or a computer program). Potential exists for the quick organization of meetings, however little research has been conducted

in this area. Available research suggests desktop VTC is effective among members already acquainted and teamed. [18, 22-23]

Audio-conferencing vs. Face-to-face Communication. Audio-conferencing removes all visual cues about other participants, reducing the ability to show understanding or agreements, forecast responses, enhance verbal descriptions, manage extended pauses, express attitudes through posture and facial expression, and provide nonverbal information. [23]

A study by France, Anderson, and Gardner shows that during AC leaders talk more. Meaning leaders dominant the conversation, where other participants of the conference may not engage in the conversation simple due to the fact the leaders have the floor. This dominance was approximately three times greater than in FTF meetings. This reduction of interaction by other participants illuminates the drawbacks to AC including group understanding, problem solving and innovation. [18, 24]

Computer Mediated Communication vs. Face-to-face Communication. CMC is typically text-based, although it can include drawings, photographs, and other images including happy faces or emoticons (types of happy faces trying to express emotion). CMC is either synchronous (i.e., chat rooms or instant messaging) or asynchronous (i.e., e-mail, discussion boards, or shared databases).

Synchronization of CMC is important to the type of information exchange taking place. When individuals or groups are in the middle of collaboration, a high degree of synchronization is important. However when information just needs to be reviewable, a high level of synchronization is not important. Figure 2.10 shows how the synchronization of different medias relates to presence of non-verbal and paraverbal cues.

In CMC all participants can contribute equally; group brainstorming is often enhanced. This enhancement of group brainstorming can be contributed to the media's wider reach for subject matter experts (i.e. reach back) and the inability of leaders to

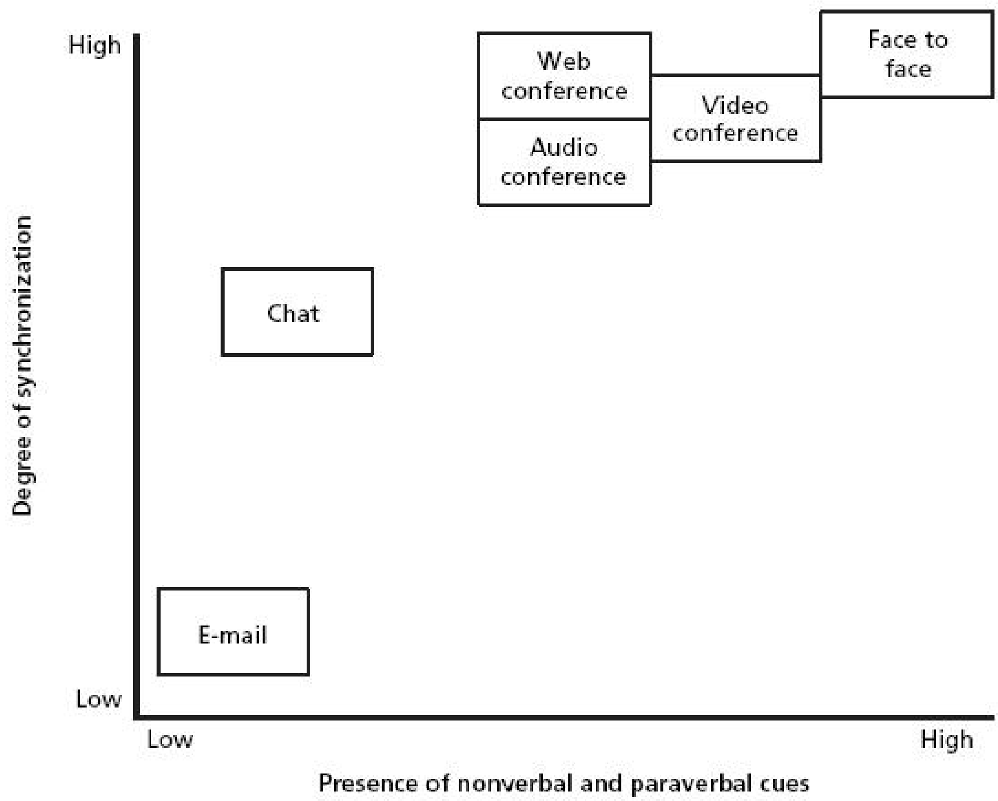


Figure 2.10 Synchronization of Different Medias [17, 5]

dominate the floor. Often in a FTF meeting, a participant may have an idea but can not express this idea until the floor is available. In a chat room ideas can be expressed freely and can be displayed as fast as the participant can type. This allows the group to focus on the task at hand rather than social considerations. [18]

However CMC groups often take longer to complete tasks than do FTF groups. [8] and miscommunication may happen more often when cues that indicate emotion are not annotated. [18]

2.2.3.4 Mitigating Problems of Mediated Communication. In review of the different communication modes one quickly realizes some tasks are better performed

by CMC than by others. Information exchange between people that evolves data that needs to be regularly reviewed can be as effective in CMC as in FTF. Again Figure 2.9 shows a notational relationship between communication media and information exchange. Take for example a weather report, if an email or webpage containing satellite pictures and text detailing predictions for hourly temperature and precipitation. CMC can often deliver the data as well as FTF. However, FTF seems best for interpersonal exchanges that require complex thinking, negotiations, or discussing problems that are ill defined. In the context of an AOC, FTF meetings are especially useful for generating and checking commitment to courses of action. [18, 65]

When thinking of split operations of an AOC, how can one best utilize and mitigate the effects of different communication modes? Many authors recommend the best way to mitigate the effects of different communication modes is to first have FTF interaction with team members and joint training on the use of different communication methods. This can be referred to as "grounding in communication".

An AFIT group project [16, 32] cites Herbert H. Clark's theory on "grounding" [17]. "In essence grounding is the process of the listener understanding, the speaker knowing the listener understood, and the listener knowing the speaker knows the listener understood. Over time, the communicators build 'common ground' - information that participants know that they all know. Large common ground becomes a significant advantage for communicators." Take for examples special operations forces. They often are able to pass significant amount of information, between members of the squad, through hand signals. This only can come with time and training.

2.2.4 Modeling Communication. The method presented by the previous AFIT group to model communication in an AOC provides a quantitative analysis that evaluates the communication media available, the communication path, and assigns values to the particular information characteristics being evaluated. To note, the communication path identifies the different domains that the information must travel through and is covered

in Section 2.2.4.1. Assigning value to the Information Characteristics may include complexity, speed, and security and is covered in Section 2.2.4.2. Section 2.2.4.3 covers evaluating the communication media and communication path with the values assigned to the Information Characteristics.

2.2.4.1 Determining Communications Paths. Determining communication paths involves identifying the different domains in which information may reside. The domains include the physical, informational, and cognitive domains. Information at the physical domain is considered the "truth" source, meaning this is the best quality of information. This may or may not be one hundred percent accurate, but it is the highest fidelity the information will have in the communication process. "Truth sources" include physical data (i.e. radar, infrared sensors), and guidance (i.e. plans and intentions). The information domain is just data. Interpreting and understanding information is the cognitive domain.

Different communication path information might take include cognitive to cognitive (C-C), physical to information to cognitive (P-I-C), cognitive to information to information to cognitive (C-I-I-C), and information to information (I-I). C-C can be considered as individuals or groups talking face-to-face. P-I-C can be considered the sensor-to-shooter path. The I-I path is simply data transfers assuming a network with sufficient bandwidth. C-I-I-C can be considered as CMC, or even VTC or AC. Figure 2.11 shows the communications paths.

2.2.4.2 Information Characteristics. When making a list of characteristics of information, the IDE group [16, 32] modified definitions from other sources to more appropriately meet the needs of the AOC. The intent of the list in Table 2.2 is to determine which communication medium would best handle a particular information characteristic. [16, 38]

After examining the list, Importance, Complexity, Speed, and Security were determined to be the best characteristic to evaluate the effectiveness of the communication

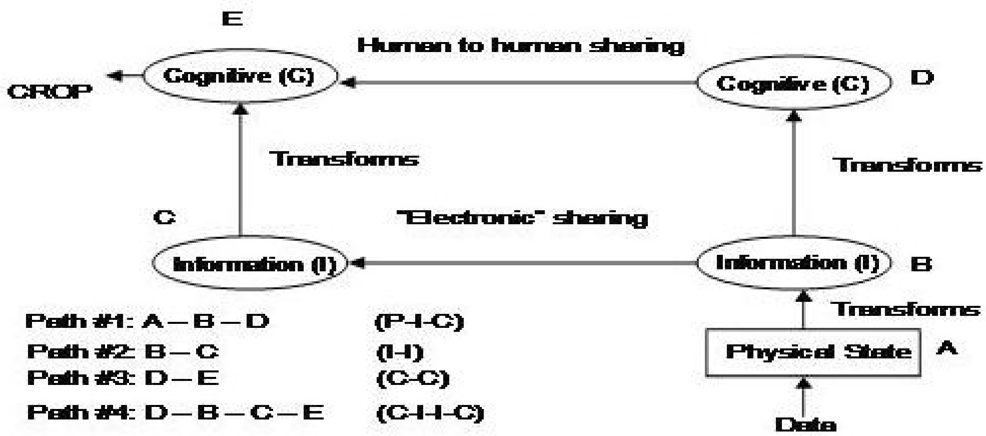


Figure 2.11 Communication Paths [16, 36]

Table 2.2 Characteristics of Information [16, 38]

Automatic	degree to which information can be input into communications network **without the need for human action** or interpretation
*Complexity	degree to which information **cannot be captured/stored in written or graphic medium**
Completeness	measures **how well** the communications network transmits relevant information
Correctness	measures the communications network transmission of the relevant information **without degradation**
Currency	measures the **rate** (time of end to end transmission refresh rate) at which the communications network transmits relevant aspects of information to each user without degradation
*Importance	measure of **criticality** of the data (BDA critical for TCT, strategy is significant, especially in planning but not critical for real time decisions)
Detail	measure of **precision** of information
*Security	measurement of **level of classification** at which the data will be transmitted and protected, and its ability to allow only specified users access
*Speed	measurement of how **fast** message needs to be understood by receiver, including confirmation back to the sender

mediums. Although important, the other characteristics were not considered as having primary effects on the success of information exchange. [16] After further evaluation of the four main characteristics, our group felt that Importance of the information could

actually be made up of Speed, Security, and Reliability of the communication medium. Speed refers to how fast important information can be created; security is how secure that information must be during transmission; Reliability refers to the dependability of the particular medium to transmit said information. In our effort to evaluate the effects of the communication medium on the ATO cycle we defined the reliability of the communication medium as a measure of how often and how long a communication medium is interrupted.

2.2.4.3 Evaluate Communication Media and Path. The ability of the communication medium and the requirements of the communication path are two areas that are difficult to measure objectively. To measure theses areas objectively a numerical system ranging from 0 to 10 (10 being best) is used to assign value to the characteristics of information. Tables 2.3 through 2.6 show the scale definitions.[16, 40-41]

Table 2.3 Ranking Communication Complexity

Complexity	
10	Impossible to communicate without explanation, meaning is important
9	Difficult to convey without listener feedback
8	Difficult to convey
7	Can be represented by text, graphic somewhat inaccurate
6	Can be represented by text, graphic somewhat accurate
5	Can be represented by text, graphic accurate
4	Easily represented by text and graphics
3	Easily represented by text or graphics
2	Easily represented by text or graphics extremely accurately
1	Compiled raw data
0	Raw data

In evaluating the limitations of the communication mediums in a particular communication domain, a combination of theory, current usages of the AOC, and normal usage is employed. Using the 0-10 scales, the limitations for each communication in the AOC are then assigned. Table 2.7 shows the values chosen for the limitations of the communication mediums. Each communication medium is assumed to operate at a certain level or below for each of the four characteristics.

Table 2.4 Ranking Communication Security

Security	
10	Highly critical that only a very few, certain parties receive, sender assured
9	Critical that only a very few, certain parties receive, sender assured
8	Critical that only a very few, certain parties receive
7	Medium distribution, top secret level
6	Large distribution, top secret level
5	Small distribution, secret level
4	Medium distribution, secret level
3	Large distribution, secret level
2	FOUO
1	Military information
0	Open

Table 2.5 Ranking Communication Speed

Speed	
10	Instantaneous
9	1 minute
8	5 minutes
7	30 minutes
6	1 hour
5	Several hours
4	1 day
3	Several days
2	1 week
1	At any time
0	No requirement

2.2.4.4 Evaluate AOC Information Exchange Requirements. The DoDAF OV-3 product from the AOC architecture created by MITRE identified over 1200 IERs from the AOC to external nodes. MITRE listed an additional 283 IERs internal to the AOC. While the OV-3 is ideal to identifying IERs, the OV-3 created by MITRE fails to identify sending and receiving parties in the AOC. [16] Table 2.8 shows an example of internal IERS identified in the OV-3. Because the current form of the OV-3 cannot be used to evaluate or model the IERs between divisions in the AOC the previous AFIT research group decided to focus on the communications involved in 51 processes performed by the five AOC divisions.

Table 2.6 Ranking Communication Reliability

Reliability	
10	Uninterruptible
9	Extremely seldom temporary losses
8	Very seldom temporary losses
7	Seldom temporary losses
6	Sporadic temporary losses
5	Extremely seldom extended losses
4	Very seldom extended losses
3	Seldom extended losses
2	Sporadic extended losses
1	Widely spaced moments of availability
0	No connectivity

Table 2.7 Limitations of Communications Media [16, 41]

	Limiting Values																		
	C-C					C-I-I-C					I-I					P-I-C			
	Complexity	Speed	Security	Reliability		Complexity	Speed	Security	Reliablity		Complexity	Speed	Security	Reliablity		Complexity	Speed	Security	Reliablity
Face to Face	10	10	10	10															
Video Teleconference	9	8	8	6															
Telephone	7	10	8	9		7	10	8	9										
Chat Room	4	7	6	7		4	7	6	4										
Email	3	5	6	7		3	5	6	4		3	5	6	4		3	5	6	4
Bulletin Board	2	1	4	6		2	1	4	5		2	1	4	5		2	1	4	5

The previous AFIT research group states, “The Senior Mentor’s briefing, ’Overview of How the C/JFACC Commands and Controls Air and Space Power through the C/JAOC,’ emphasized the processes within the AOC. This briefing, developed by the C2 Warrior School at the 505th TRS, used the processes as outlined in the AFDD-2 to illustrate the responsibilities each division has during the AOC cycle.” [16, 18] With use of the 505th courseware, the group examined each of the 51 processes to determine the high level IERs

Using a similar methodology of evaluating the limitations of communication mediums, the required communication medium for each IER could be determined. Each IER is assigned into one of the four communication paths (C-C, P-I-C, C-I-I-C, I-I), and

Table 2.8 MITRE OV-3 Excerpt

NEED LINE	INFORMATION EXCHANGE	SENDING NODE	SENDING ACTIVITY	RECEIVING NODE	RECEIVING ACTIVITY
CAOC Internal	Mission Analysis	CAOC	CAOC-1.1-Analyze Mission CAOC-1.1.1-Analyze CINC/JFC Mission and Intent CAOC-1.1.1.6-Analyze Command Relationships	CAOC	CAOC-1.2-Determine Aerospace Objectives CAOC-1.2.1-Review Component (if available) and JFC Objectives
CAOC Internal	Air and Space Objectives and Associated Tasks	CAOC	CAOC-1.2-Determine Aerospace Objectives CAOC-1.2.3-Assign Tasks (including ISR) to Aerospace Objectives	CAOC	CAOC-1.3-Develop Strategy/Courses of Action CAOC-1.3.1-Prioritize Objectives CAOC-1.3.9.1-Translate air objectives into target sets
CAOC Internal	Courses of Action (COA)	CAOC	CAOC-1.3-Develop Strategy/Courses of Action CAOC-1.3.Z10-Determine Phasing	CAOC	CAOC-1.4-Compare Aerospace Courses of Action (COA) CAOC-1.4.1-ID Standards of Comparison
CAOC Internal	Courses of Action (COA) Analysis	CAOC	CAOC-1.4-Compare Aerospace Courses of Action (COA) CAOC-1.4.4-Compare against adversary COA(s)	CAOC	CAOC-1.5-Select Strategy/COA
CAOC Internal	Air and Space COA Nomination	CAOC	CAOC-1.5-Select Strategy/COA	CAOC	CAOC-1.6-Develop Joint Air Operations Plan (JAOP) CAOC-1.6.1-Finalize Aerospace COA CAOC-1.6.1.1-Obtain JFACC approval of Aerospace
CAOC Internal	JAOP Draft	CAOC	CAOC-1.6-Develop Joint Air Operations Plan (JAOP) CAOC-1.6.5-Finalize Detailed Joint Aerospace Operations Plan	CAOC	CAOC-1.7-Obtain Approval of Detailed Joint Aerospace Operations Plan CAOC-1.7.1-Obtain JFACC approval
CAOC Internal	Air and Space Objectives Input	CAOC	CAOC-1.2.2-Identify potential (overarching) aerospace objectives CAOC-1.2.1-Review Component (if available) and JFC Objectives	CAOC	CAOC-1.2.3-Assign Tasks (including ISR) to Aerospace Objectives CAOC-1.2.2-Identify potential (overarching) aerospace objectives

ascribed relevant values to the data and communication requirements in accordance with the 0-10 scales. Then to compare the limitations of the communication medium and the communication requirements needed for the IERs the AFIT research group created what they call the OV-3x. Table 2.9 show the OV-3x created by the group to evaluate the effects of the communication mediums on the ATO cycle for a split AOC. Appendix B section OV-3 shows our efforts to use the OV-3x and map each of the 51 processes to appropriate transitions used in our executable architecture. Sections 3.2 and 4.1 explain our efforts of using the OV-3x in an executable architecture.

2.3 DoD Architecture Framework (DoDAF)

For distributed AOC operations, we must carefully plan out which form of communication is going to be used for each step of the ATO cycle. The tool used to communicate this design is architecture. This is similar in importance to a blue-print used

Table 2.9 OV-3x [16, 47]

			DOMAIN TRANSFER				COMM REQTS				ACCEPTABLE MEDIUMS					
	Party 1	Party 2	C-C	C-I-C	I-I	P-I-C	Reliability	Complexity	Speed	Security	FTF	VTC	Telephone	Chat Room	Email	Bulletin Board
AIR MOBILITY DIVISION																
Integrates, directs execution																
	AMCT	FACC, AOC director	x				10	6	5	3	x					
	AME	TACC	x				10	6	6	3	x					
Maintains flow of assets																
	ARCT	Wing AOCs														
	AECT	Combat Plans	x				6	3	3	1	x	x	x	x	x	
	ALCT	Air Attache		x			6	3	2	3			x	x	x	
	ALCT	Aerial Port	x				6	3	2	2	x	x	x	x	x	
Coordinates air mobility support																
	AMCT	DIRMOBFOR	x				5	6	5	3	x	x	x			
	AMCT	Wing AOCs	x				5	4	3	4	x	x	x	x		
	AMCT	Coalition	x				5	7	3	4	x	x	x			
	AMCT	TALCE		x			6	5	3	4			x			

to build a house. An architecture shows how everything fits together. The structure of the underlying components is defined and the communications between those components are depicted.

The DoDAF defines a common approach in the DoD for architecture description, development, presentation, and integration for both warfighting operations and business operations and processes. [7, 1-1] Easier said, the DoDAF gives the DoD a common language to describe architectures. This section will cover what an architecture is, the history of the development of the DoDAF and DoDAF's description and definition of different architecture views. Views used in the development of executable architectures will be covered in more detail in section 2.4.

2.3.1 Architectures and the DoD. This section details why the DoD decided to start using architectures and what an architecture is. In October 1995, the Deputy Sectary of Defense directed that a DoD-wide effort be undertaken to define and develop better means and processes for ensuring that Command, Control, Communications, and Intelligence (C4ISR) capabilities meet the needs of the warfighter. [6, 1-5] This was in direct response to increasing joint and multinational operations being conducted by the DoD. The DoD had discovered that newer technology was being applied piecemeal. [29, 14] The intra-service communication systems could not correspond with or work with

communication systems from other services let alone communication systems of other countries. For example, during the Persian Gulf War of 1991, it became evident that communication between Services was severely lacking. The ATO generated at the AOC by Air Force personnel could not be transmitted securely to Navy aircraft carriers. In an effort to circumvent this problem the AOC would generate a paper copy of the ATO and have it flown out to the Navy's aircraft carriers. [29, 14]

With this inability of the communication systems to correspond to one another and with increasing uncertainty about requirements, rapid changes in technology, changes in organizational structures, and a widening spectrum of missions and operations, the DoD had to find a way to deal with these uncertainties. [10, 1] In the past, the services, individual commands, and independent DoD agencies would develop their own solutions to these uncertainties. [25, 1] However, the DoD had to deal not only with these uncertainties, but also their other interoperability issues. In need of a solution, the DoD looked to information architectures. Dr. Alexander Levis, former Chief Scientist of the Air Force and a proponent of executable architectures, states "Information architectures can provide current and future descriptions of a domain composed of components and their interconnections, actions or activates those components perform, and rules or constraints for those activates." [10, 1] Hence, information architectures can provide the DoD a way to define and develop C4ISR capabilities to meet the needs of the warfighter.

2.3.1.1 Description of Architecture. What is an architecture? Often when one thinks of an architect and architectures, one thinks of a designer and the design of buildings and bridges. However, in the world of Systems Engineering we think of an architect as one needed to create any structure that is both unprecedented and complex. [21] [22] According to IEEE an Architecture is defined as:

> The fundamental organization of a system embodied in its components, their relationships to each other and to the environment and the principles guiding its design and evolution. [3]

According to Rechtin and Levis, “it is possible to derive several characteristics of systems architects and the architectures they produce. The first is that an architecture is needed only if the system is unprecedented and complex. Second the architect is driven by the special needs of the customer and tends to develop the architecture in a top-down manner.”[10] This is very useful for the DoD, in fact, because the needs, requirements, and the organizational hierarchy of the DoD are unique. The third characteristic of a systems architect and the architectures produced is that the task of the architect is to elicit those needs and to produce a “description” that can demonstrate to the customer that the system to be produced (in conformance with the architecture) will satisfy the customer’s wants and needs. This means the architecture does not include the details of the final designs. [10] The architecture is a design of what the system is supposed to accomplish versus how the system will actually accomplish it; the architecture is a static model of the system. If an executable architecture could be created from a static architecture then the architect could show the user how the system is to accomplish its mission in a dynamic form.

2.3.1.2 Structured Analysis of Architectures. An Architecture is defined above as “the fundamental organization of a system embodied in its components, their relationships to each other and to the environment and the principles guiding its design and evolution.” So, what does this mean? It means an architecture is the views of the components of a system on how they relate to one another and react with their environment. It is necessary to describe the components, of these views, that will implement the system. The components include: the hardware, software, personnel, and facilities that will comprise a system and perform the processes. [10, 228] These architectural views can be different because people can view a system from different perspectives. Such a pilot views his airplane differently than the engineer that designed it.

These views may be broken up into three types; a *Functional*, *Physical*, and *Technical* view. These views can be presented in graphical, textual or tabular form. Structured Analysis of architectures is the process of decomposing these views of a system with the appropriate presentation of components.

Using Structured Analysis in a basic systems engineering approach, an architecture is composed of two constructs: the *Functional Architecture* (view) and the *Physical Architecture* (view). According to Levis, "The *Functional Architecture* is a set of activities or functions, arranged, in a specified partial order that, when activated, achieves a set of requirements. Similarly, a *Physical Architecture* is a representation of the physical resources, expressed as nodes that constitute the systems and their connectivity, expressed in the form of links." [10] The *Technical Architecture View* while not part of the Structured Analysis approach is defined as the minimal set of rules governing the arrangement, interaction and interdependence of system parts and elements. [10][6] The implementation of these views in the DoD will be covered in Section 2.3.2 Figure 2.12 below shows Levis' three phases of architecture development.

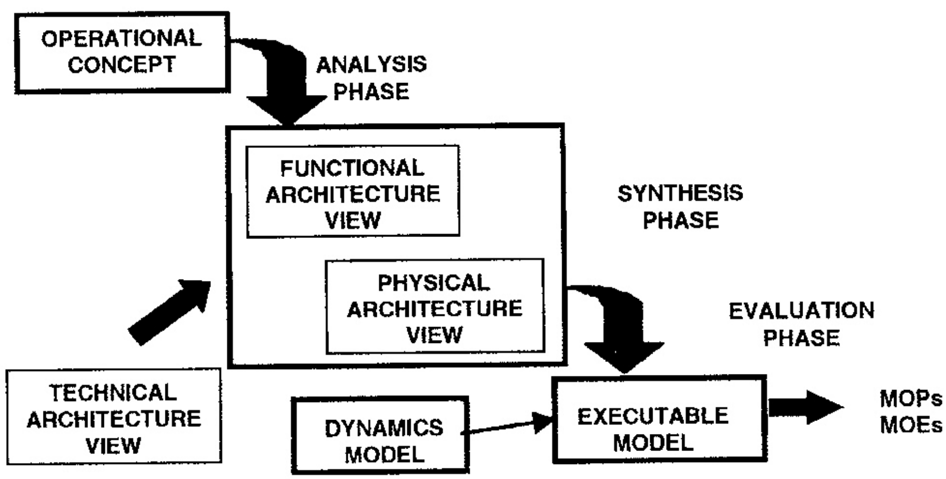


Figure 2.12 Dr. Levis' Basic Architecture Design Process

2.3.1.3 History of DoDAF Development. DoD realized the need for common approach for describing architectures in the mid-1990s. Up to that time, the individual Commands, Services, and Agencies traditionally described their architectures using their own techniques, vocabularies, and presentation schemes. [6, 1-5]

In October 1995, the Deputy Secretary of Defense directed that a DoD-wide effort be undertaken to define and develop better means and processes for ensuring that C4ISR capabilities meet the needs of the warfighter. A C4ISR Integration Task Force (ITF) was established in response to that direction. The Integrated Architectures Panel (IAP), one of several panels established by the ITF, published the C4ISR Architecture Framework, Version 1 on 7 June 1996. [6, 1-5]

The C4ISR Architecture Working Group was established in October 1996, to continue the work of the IAP. Their effort resulted in the C4ISR Architecture Framework Version 2.0, dated 18 December 1997. In February 1998, the Architecture Coordination Council made up of the Undersecretary of Defense for Acquisition and Technology (USD[A&T]), the Assistant Undersecretary of Defense for Command , Control, Communication and Intelligence, and the Joint Staff/J6, published a memorandum mandating the use of Version 2 for all C4ISR architecture descriptions. [6, 1-6]

Due to the fact that DoD Policy encouraged the use of Architectures, the C4ISR Architecture Framework Version 2 was transformed into the DoDAF in 2003. To accomplish this evolution the Architecture Framework Working Group (AFWG) was established by the DoD Chief Information Officer. The AFWG placed under the direction of the Director of Architectures and Interoperability. The working group was composed of representatives of the Joint Staff, Military Services, and other DoD components. [6, 1-6]

2.3.2 DoDAF Views and Products. As discussed before architectures can be presented in three views, a *Functional*, *Physical*, and *Technical* view. However, in the DoDAF, the Functional view is called the Operational View and the Physical View is referred to as the Systems View. The DoDAF defines a standard set of products for the definition of an integrated architecture. It begins by defining three views, each of which contains a number of products: the Operational View (OV), Systems View (SV), and the Technical Standards View (TV). [6, 1-2] Additionally, the All- View (AV) set of products

captures overarching aspects that set the scope and context of the architecture. [6, 1-3] Table 2.10 briefly describes the 26 products, which are defined in the DoDAF.

The OV consists of nine products, which collectively describe, "the tasks and activities, operational elements, and information required to accomplish DoD missions." The SV is a set of 13 "graphical and textual products that describes systems and interconnections providing for, or supporting, DoD functions." The SV products assign systems to perform the functions laid out in the OV products. Finally, the two TV products describe "the minimal set of rules governing the arrangement, interaction, and interdependence of systems parts or elements." [6, 1-3] Typically, the TV products specify industry or other technical standards as well as adopted conventions and rules that govern the communication between the systems described in the SV products.

Given these product definitions, an integrated architecture is an architecture description in which the "data elements defined in one view are the same as architecture data elements referenced in another view." In other words, the architecture products of an integrated architecture display the property of concordance, such that there is direct traceability from data elements in OV products to SV products, and from SV products to TV products. Figure 2.13 illustrates the traceability between products. Chapter 7 of the DoDAF goes into more detail on how data elements from one product may relate to other products.

Not all 26 products are required to form an integrated architecture. DoDAF Vol 1, defines the minimum set as: AV-1, AV-2, OV-2, OV-3, OV-5, SV-1, and TV-1. Further products should be developed as the intended use of the architecture requires. The key is that the architecture be specified at a level of granularity appropriate to the complexity of the system and the intended use of the architecture.

2.3.2.1 Product Descriptions. As previously discussed, integrated architectures encompass products from all three views. Certain architecture products from each view are more essential than others, particularly in the discussion of the ATO Cycle.

Table 2.10 Architecture Products Description [6, 2-4]

Applicable View	Framework Product	Framework Product Name	General Description
All Views	AV-1	Overview and Summary Information	Scope, purpose, intended users, environment depicted, analytical findings
All Views	AV-2	Integrated Dictionary	Architecture data repository with definitions of all terms used in all products
Operational	OV-1	High-Level Operational Concept Graphic	High-level graphical/textual description of operational concept
Operational	OV-2	Operational Node Connectivity Description	Operational nodes, connectivity, and information exchange needlines between nodes
Operational	OV-3	Operational Information Exchange Matrix	Information exchanged between nodes and the relevant attributes of that exchange
Operational	OV-4	Organizational Relationships Chart	Organizational, role, or other relationships among organizations
Operational	OV-5	Operational Activity Model	Capabilities, operational activities, relationships among activities, inputs, and outputs; overlays can show cost, performing nodes, or other pertinent information
Operational	OV-6a	Operational Rules Model	One of three products used to describe operational activity—identifies business rules that constrain operation
Operational	OV-6b	Operational State Transition Description	One of three products used to describe operational activity—identifies business process responses to events
Operational	OV-6c	Operational Event-Trace Description	One of three products used to describe operational activity—traces actions in a scenario or sequence of events
Operational	OV-7	Logical Data Model	Documentation of the system data requirements and structural business process rules of the Operational View
Systems	SV-1	Systems Interface Description	Identification of systems nodes, systems, and system items and their interconnections, within and between nodes
Systems	SV-2	Systems Communications Description	Systems nodes, systems, and system items, and their related communications lay-downs
Systems	SV-3	Systems-Systems Matrix	Relationships among systems in a given architecture; can be designed to show relationships of interest, e.g., system-type interfaces, planned vs. existing interfaces, etc.
Systems	SV-4	Systems Functionality Description	Functions performed by systems and the system data flows among system functions
Systems	SV-5	Operational Activity to Systems Function Traceability Matrix	Mapping of systems back to capabilities or of system functions back to operational activities
Systems	SV-6	Systems Data Exchange Matrix	Provides details of system data elements being exchanged between systems and the attributes of that exchange
Systems	SV-7	Systems Performance Parameters Matrix	Performance characteristics of Systems View elements for the appropriate time frame(s)
Systems	SV-8	Systems Evolution Description	Planned incremental steps toward migrating a suite of systems to a more efficient suite, or toward evolving a current system to a future implementation
Systems	SV-9	Systems Technology Forecast	Emerging technologies and software/hardware products that are expected to be available in a given set of time frames and that will affect future development of the architecture
Systems	SV-10a	Systems Rules Model	One of three products used to describe system functionality—identifies constraints that are imposed on systems functionality due to some aspect of systems design or implementation
Systems	SV-10b	Systems State Transition Description	One of three products used to describe system functionality—identifies responses of a system to events
Systems	SV-10c	Systems Event-Trace Description	One of three products used to describe system functionality—identifies system-specific refinements of critical sequences of events described in the Operational View
Systems	SV-11	Physical Schema	Physical implementation of the Logical Data Model entities, e.g., message formats, file structures, physical schema
Technical	TV-1	Technical Standards Profile	Listing of standards that apply to Systems View elements in a given architecture
Technical	TV-2	Technical Standards Forecast	Description of emerging standards and potential impact on current Systems View elements, within a set of time frames

The OV-5, OV-6a, OV-6b, and OV-7 products will be covered in more detail because these are the products that were utilized to create our CPN for the ATO Cycle. All the

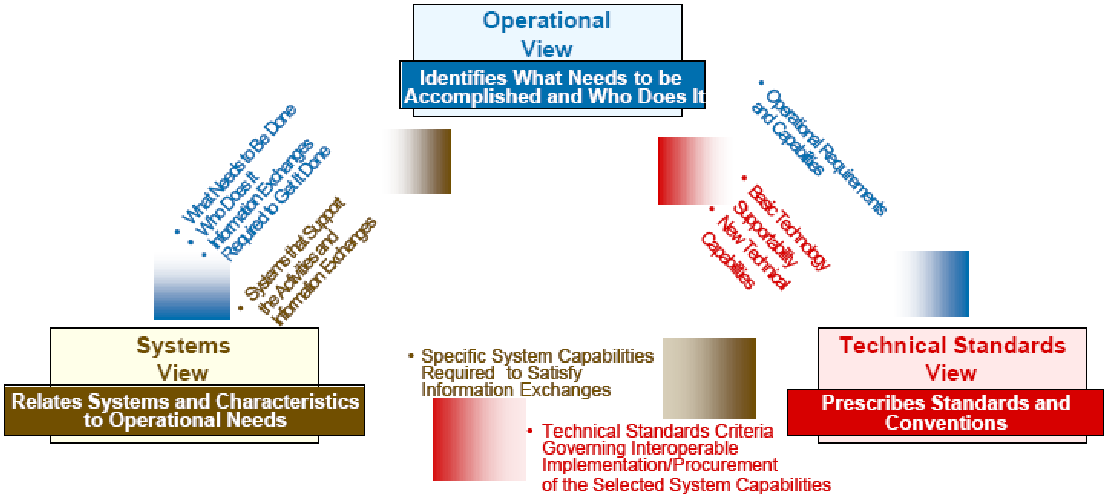


Figure 2.13 Linkages Between DoDAF Views [7, 2-1]

architecture products generated, using the DoDAF, for the ATO Cycle, can be seen in APPENDIX B. From this point on, when the word architecture is used it will be referring to integrated architectures.

There are two AV products, the Overview and Summary Information (AV-1) and the Integrated Dictionary (AV-2). The AV-1, "provides executive-level summary information in a consistent form that allows quick reference and comparison among architectures." [7, 3-1] The AV-1 has two purposes. In the initial phases of developing architecture products for a system; the AV-1 serves as a planning guide that identifies the need for the architecture, the viewpoint from which the architecture is developed and the context. The context can include such things as mission, doctrine, relevant goals, and vision statements, concept of operations, scenarios, and information assurance. [7, 3-10] The second purpose of the AV-1 is to provide summary textual information concerning the architecture upon completion. [7, 3-1]

The AV-2 is the integrated dictionary and is sometimes referred to as the data dictionary. Like a dictionary, the AV-2 defines all architectural elements and common terms to prevent misconceptions. The architecture data in the products with graphical representations consists of each labeled item. These labeled items include ICOMS

(Inputs, Controls, Outputs, and Mechanisms), Entities, Activities, and Transitions. The products with textual descriptions also have labeled items that are listed in the AV-2. [7, 3-9] Architects should use standard terms or vocabulary. Often in the DoD specific communities have their own vocabulary that may be used differently in other operational communities. Take for example, “the use of the term *track* refers to very different concepts in the carrier battle group community than in the mine-sweeper community. Yet both of these communities are Navy operational groups and may participate together in littoral warfare task forces.” [7, 3-11]

The High Level Operational Concept Graphic or OV-1 is simply a graphic that represents a Concept of Operations. This graphic is often just a presentation slide illustrating what the architecture is supposed to do, and how it is supposed to do it. [7, 4-1]

The OV-2 is the Operational Node Connectivity Description Diagram. The OV-2 graphically depicts the exchange of information between nodes using a needline. Nodes can be both internal and external to a system. The DoDAF defines a node as, “the element of the operational architecture that produces, consumes, or processes information” and a need line, “documents the requirement to exchange information.” [7, 4-7] The need line does not indicate how the information exchange is accomplished just that it needs to be done. Just to note a single need line in an OV may imply multiple forms of communications. Information exchanges and their modes of transfer will be covered more in depth in the Communication Theory section.

The Operational Information Exchange Matrix (OV-3) identifies, “who exchanges what information, with whom, why the information is necessary, and how the information exchange occur.” [7, 4-16] The OV-3 is the ideal product for the scope of out project however, the current OV-3 products available for the AOC are not detailed enough. This will be covered in more detail in the AOC architecture section and the Communication Theory section.

The Organizational Relationship Chart (OV-4), is simply an organizational chart often presented in a hierarchical tree format. The OV-4 can illustrate the relationships among organizations or resources in an architecture. These relationships can include command and control relationships, command-subordinate relationships, and coordination relationships between equals. [7, 4-27]

The Operational Activity Diagram (OV-5), describes the activities or functions that occur in a system. The OV-5, "describes capabilities, operational activities (or tasks), input and output (I/O) flows between activities, and the I/O flows to/from activities that are outside the scope of the architecture." [7, 4-31] The OV-5 is often created by using the Integration Definition for Function Modeling (IDEF0).

In IDEF0, a function is a transformation that turns inputs into outputs. Inputs to be transformed into outputs enter a "function box" from the left, controls that guide the transformation process enter the top, mechanisms (physical resources that perform the function) enter the bottom, and outputs leave the right. [12, 67] IDEF0 uses ICOM arrows to represent the Inputs, Controls, Outputs, and Mechanisms. DoDAF suggests that controls should represent Doctrine, while mechanisms should represent organizations or operational nodes.

Like many of the diagrams in the DoDAF, the OV-5 can be decomposed. For example, "the top level activity (called A0 in IDEF0) alone would make up the "context diagram." It may then be decomposed into three lower level (or "detailed level") activities, A1, A2, and A3, which are depicted in another diagram. A1, A2, and A3 are a decomposition of A0. Figure 2.14 shows a decomposed IDEF0 diagram; inputs come in from the right (I1, I2), outputs exit from the left (O1, O2, O3), controls come in from the top (C1, C2, C3), and mechanisms come in from the bottom (M1).

The decomposition levels of the OV-5 should be aligned with the operational nodes that are responsible for conducting the operational activities represented in the OV-2. [7, 4-33] In the case of this study, the OV-5 was decomposed to the level of the operational divisions in the AOC. This will be covered more in depth in Section 2.5.1.

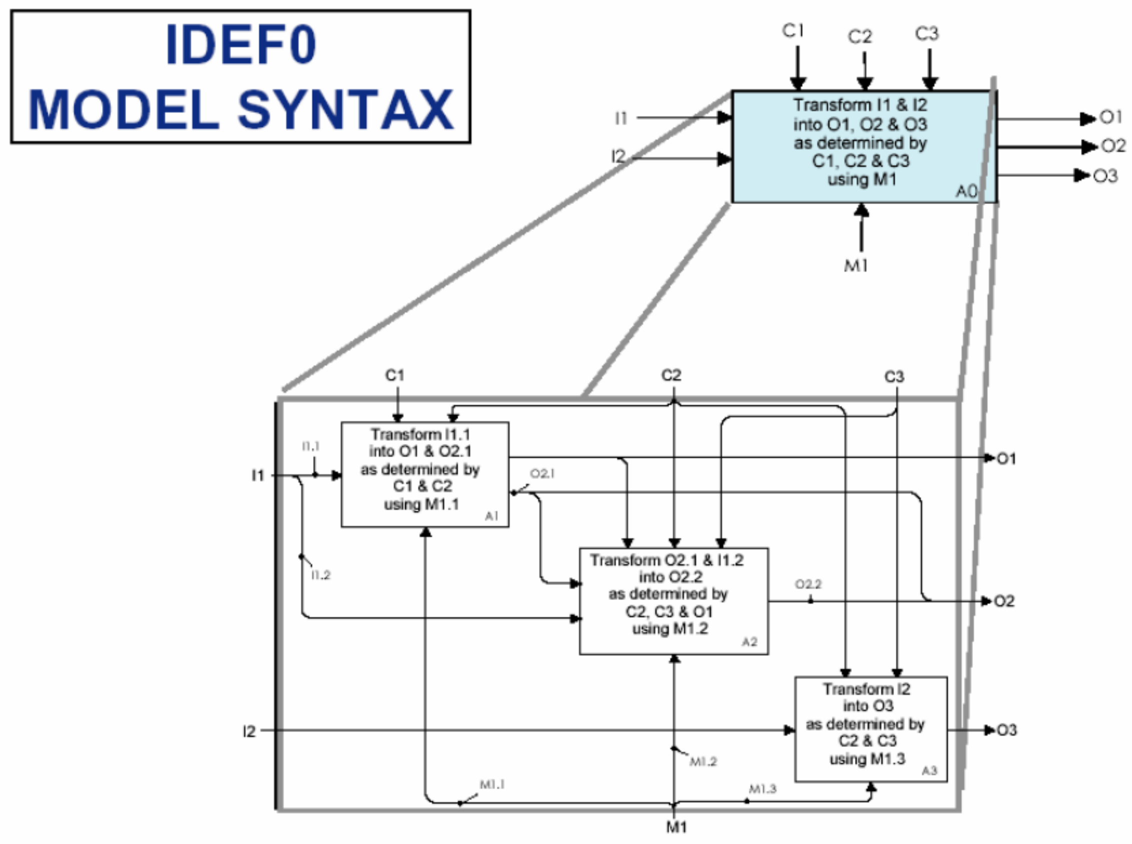


Figure 2.14 Functional Decomposition in IDEF0 [12, 71]

DoDAF refers to the OV-6 as the Operational Activity Sequence and Timing Descriptions. The OV-6 is broken up in to three separate descriptions.

- The Operational Rules Model (OV-6a)
- Operational State Transition Diagram (OV-6b)
- Operational Event-Trace Description (OV-6c)

The OV-6 products model the dynamic behavior of an operational system. Where the previous OV products modeled the "static architecture" of the system, the OV-6 products are concerned with the timing and sequencing of the events of the activities in the OV-5. [7, 4-43]

The OV-6a uses Logical Rules, such as IF-THEN-ELSE statements, to define or constrain some aspect of an activity. “These rules might prescribe the specific set of inputs required to produce a given output.” [7, 4-45] For example, in the ATO cycle before the Combat Plans Division can produce the Joint Integrated Prioritized Targeting List (JIPTL), they must receive guidance from Air Operations Directive (AOD) and Battle Damage Assessment from the previous day’s missions.

The OV-6b is a graphical method of describing how an operational node or activity responds to various events by changing its state. The DoDAF states, “The (OV-6b) relates states, events, and actions. A state and its associated action(s) specify the response of an operational activity to events. When an event occurs, the next state may vary depending on the current state (and its associated event), the event, and the rule set or guard conditions. A change of state is called a *transition*. Each transition specifies the response based on a specific event and the current state. Actions may be associated with a given state or with the transitions between states.” [7, 4-50]

Figure 2.15 shows an example of a state transition diagram. The figure is taken from presentation slides created by Dr. Levis. The example is of the possible states of a defensive counter air patrol mission.

The Operational Event Trace Description (OV-6c) allows the tracing of actions in a particular scenario or critical sequence of events of an operational thread. [7, 4-55] The OV-6c is often referred to as the sequence diagram in the Unified Modeling Language (UML).

The Logical Data Model (OV-7) is the final OV product. The OV-7, “defines the architectures domain’s system data types (or entities) and the relationships among the system data types.” [7, 4-62] The DoDAF continues to state, “The OV-7 defines each kind of system data type associated with the architecture domain, mission, or business, as its own entity, with its associated attributes and relationships. These entity definitions correlate to OV-3 information elements and OV-5 inputs, outputs, and controls”

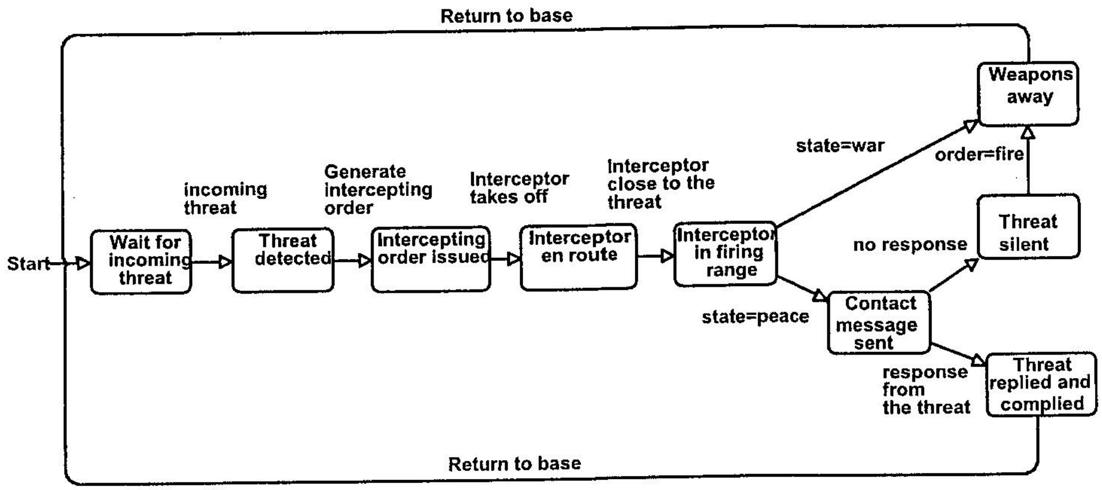


Figure 2.15 High Level State Transition Diagram

The OV-7 can be modeled with either the Integrated Definition for Data modeling or in UML where it is referred to as a Class Diagram. Figure 2.16 from DODAF shows an Example.

The 13 Systems Views (SV) products is a set of, “graphical and textual products that describes systems and interconnections providing for, or supporting, DoD functions.” [7, 5-1] The SV products assign systems to perform the functions laid out in the OV products. Only the SV-1, SV-4, and SV-5 will be covered here. For more information, see the DoDAF.

The Systems Interface Description (SV-1), “depicts systems nodes and the systems resident at these nodes to support organizations/human roles represented by operational nodes of the Operational Node Connectivity Description (OV-2).” ([7, 5-1] The SV-1 links the OV and SV products by showing the system nodes that support the operational nodes.

The Systems Functionality Description (SV-4), “describes system functions and the flow of system data among system functions.” [7, 5-25] There is a correlation between the OV-5 and SV-4. However, there may not be a one-to-one mapping of system functions to

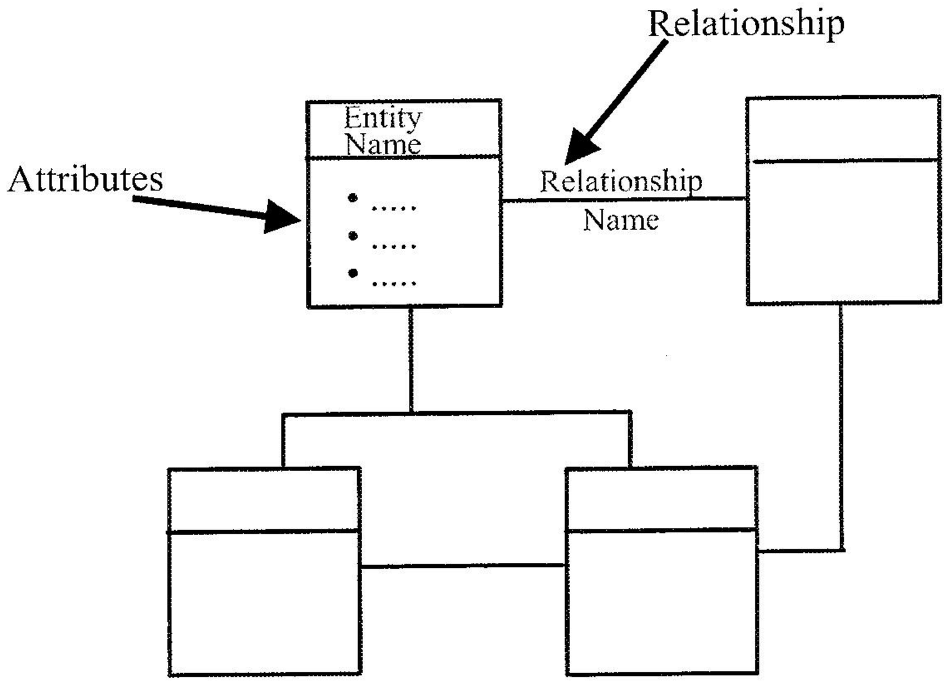


Figure 2.16 OV-7 as a UML Class Diagram [7, 4-63]

operational functions. This mapping is done in the SV-5. The Data Flow Diagram (DFD) is often the tool of choice to represent the SV-4. Figure 2.17 shows an example of a DFD.

The Operational Activity to systems Function Traceability Matrix (SV-5), “depicts the mapping of operational activities to system functions and this identifies the transformation of an operational need into a purposeful action performed by a system.” [7, 5-35] In other words, the SV-5 shows what systems are used and how they are linked together to perform an activity by a particular organization. An example of this would be how the combat plans division, in the AOC, uses the Combat Target Planning System and the electronic Joint Munitions Effectiveness Manual to evaluate the desired effects of a weapon against a particular target. The DoDAF also says, “The relationship between operational activities and system functions can also be expected to be many-to-many (i.e., one operational activity may be supported by multiple system functions, and one system

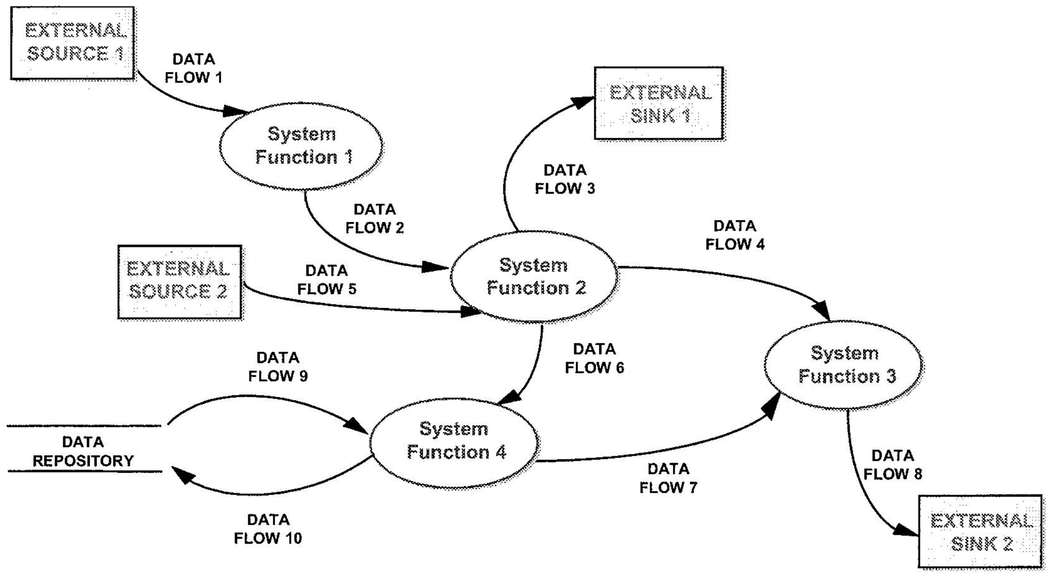


Figure 2.17 SV-4 as a Data Flow Diagram [7, 5-26]

function may support multiple operational activities)". [7, 5-35] The SV-5 of the AOC ATO cycle can be seen in APPENDIX B .

The Technical Views consist of two products. The TV, "provide the technical systems-implementation standards upon which engineering specifications are based, common building blocks are established, and product lines are developed." [7, 6-1]

The Technical Standards Profile (TV-1), "is a collection of standards that are relevant to the architecture. For military systems, the Joint Technical Architecture (JTA) is often cited in the TV-1." [29, 29] The DoDAF states, "The standards are referenced as relationships to the systems, systems functions, system data, hardware/software items or communications protocols in the SV-1, SV-2, SV-4, SV-6, OV-7 and SV-11 products." [7, 6-20]

The Technical Standards Forecast (TV-2) lists how expected standards in the TV-1 product are likely to change in the future.

2.3.3 DoDAF Architecture Description Process. While not mandated by the DoDAF, the Vol 3 Deskbook discusses a six-step step process to build the architecture descriptions and how to use architectures. Figure 2.18 depicts the six-step process. This six-step process is a generic process and the DoDAF suggests that specific organizations should tailor the process to their needs and purpose. The six-steps to the DoDAF's Generic process are listed below. [6, 5-4]

1. Determine the intended use of the Architecture
2. Determine the architecture description's scope, context, environment, and any other assumptions to be considered.
3. Based on the intended use and scope, determine what information the architecture description needs to capture.
4. Determine products to be built
5. Gather the architecture data and build the requisite products.
6. Use the architecture description for the intended purpose.

The architecture descriptions should have been built in mind for a specific purpose, that being either to support investment decisions, requirements identification, system acquisition, interoperability evaluation, operations assessment, or other purposes. [6, 5-6] The DoDAF states, "The architecture descriptions facilitate and enable these purposes but does not provide conclusions or answers. For that, human and possibly automated analysis must be applied." [6, 5-6] It is the hope of this group that the research conducted with CPN and ARENA on the ATO cycle will help to develop automated analysis techniques for architectures in other organizations for the DoD.

2.3.4 DoDAF Uses. The DoDAF and the use of integrated architectures will be key elements of the Department of Defense's transition from a threat-based force to a capability-based force. [6, 3-1] The DoD has made a long-term commitment to the development and maintenance of integrated architectures to both document

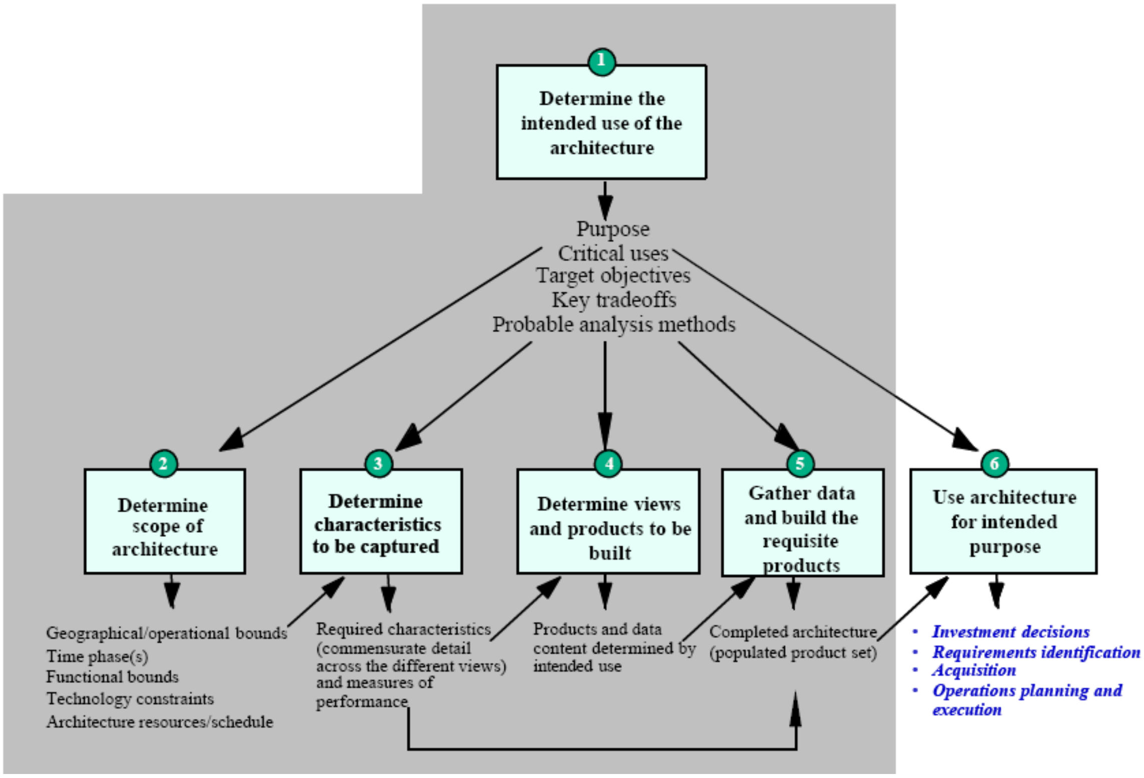


Figure 2.18 The Six Step DoDAF Architecture Development Process [6, 5-5]

current capabilities and to define requirements for future capabilities. The central role architectures are to play in all DoD acquisitions is highlighted by their central placement in the Acquisition Process defined in DODI 5000.2 shown in Figure 2.19. As these architectures become more robust and widespread they will absorb existing Capstone Requirements Documents. [5, 3]

DoDAF architecture products will be used for a variety of decision-making circumstances, ranging from capability analysis to operational planning to investment decision making. Each of the users has specific areas of interest, and if they are developing the architecture, they will focus on their areas of expertise. Ideally, a single fully specified integrated architecture will be used by all interested parties, with each extracting the data elements relevant to the decision they must make. Figure 2.20 represents a sample of potential uses for integrated architectures depending on the decision maker.

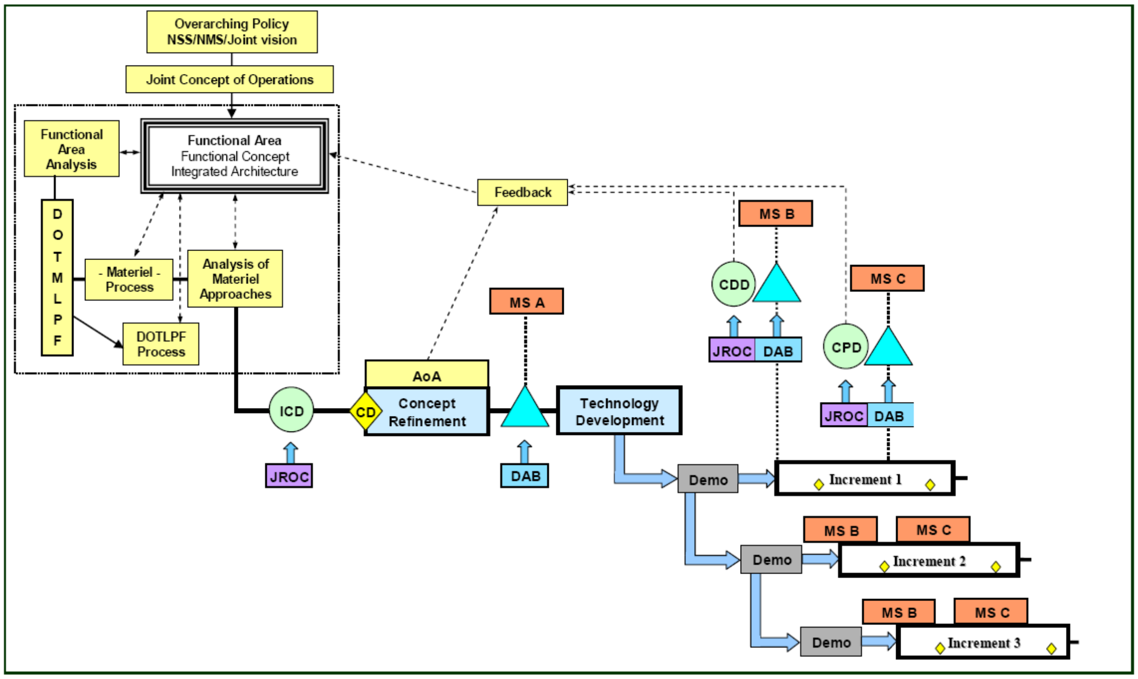


Figure 2.19 DoDI 5000.2 Acquisition Process [4, 3]

As DoDAF states, "as DoD enters into an era of Net-Centric Operations and Warfare, the ability to portray and understand complex many-to-many relationships becomes even more important." Architectures are a key tool in portraying and managing this growing complexity. One of the most promising is the potential to combine an integrated architecture with modeling and simulation tools in order to form an "executable architecture." It is possible "to show complex, dynamic organizational interactions that cannot be identified or properly understood using static models." [6, 4-50] This approach is strongly advocated by Dr. Levis. He notes, however, that the current DoDAF lacks guidance as to how executable architectures are to be developed. This Thesis attempts to answer this question and will be covered in more detail in section 2.4.

2.3.5 AOC Architecture. General Jumper (at that time Commander, Air Combat Command, now Chief of Staff, United States Air Force) declared that the AOC should be treated as a weapons system. This was in an effort to combat the ad-hoc nature of the AOC. Before this declaration, each theater command (i.e. PACAF, USAFE, CENTAF)

Increasing Breadth of Inter-operability, Integration, and Investment Concerns

Community of Interest	Typical Perspective	Representative Architectural Insights Desired
⑤ OSD Principal Staff Assistants	DoD-wide Interoperability & POM Investment Decisions	Why do the DoD enterprises (Services, Agencies) need to build or procure the systems they are proposing? Can other DoD capabilities be leveraged to perform the same functions? Can any of the proposed systems be adapted/modified to meet broader DoD interests? Are the proposed systems compliant with GIG and other directives? "How are processes simplified and work unified across DoD organizations?"
④ Joint Staff	Cross-Command Interoperability & JROC/JWCA Investment Decisions	Do the systems proposed by the various enterprises interoperate under any JTF scenario and integrate into any theater? Do these systems add to overall Joint Warfighting effectiveness without unnecessary duplication of functionality?
③ Combatant Commander	Joint Warfighting Capability & CC Investment Strategies	Do the systems proposed by the various enterprises interoperate under JTF operations postulated for this theater, and integrate into this theater's infrastructure? Do they enhance C4ISR support to Joint force operations?
② Service or Agency Enterprise	Enterprise Operations, Fit, and Conformance & Enterprise Investment Decisions	Do the systems proposed by system developers conform with this enterprise's strategic direction? with the enterprise technical architecture? with the enterprise's rightsizing policy? How are processes simplified and work unified across the Service or Agency Enterprise?
① System Developer	ICD/CDD/CPD Satisfaction & Acquisition Strategy	Does the capability or system under consideration satisfy the target MNS/ORD? Is the proposed design cost-effective in context with alternative acquisition strategies?

Figure 2.20 Value of Architectures to Different Communities [6, 3-8]

would build their own AOCs. Each individual command or organization would create and manage their own AOCs according to various local methods and designs. The first attempt to create an architecture for AOC weapon system was designated the Air Force Aerospace Operations Center Weapon System AN/USQ-163, Falconer Architecture, Block 10.1. (Short Name: AOC WS 10.1 Architecture) The Air Force contracted the MITRE Corporation of Hampton, Virginia to architect a baseline architecture for the AOC. According to Zinn, the AV-1 of the first 10.1 architecture release states, "the reference material for the architecture exists as a collection of mostly unrelated heaps of PowerPoint

briefings, documents, and emails. It is inconsistent, sparse in many areas, out of date, uneven in fidelity, or in some cases non-existent." [29, 37]

The AOC WS 10.1 Architecture that was used for our study was released by MITRE January 11, 2004. According to the AV-1, "this architecture reflects baseline portions of the Al-Udeid Air Base (AUAB) AOC, and incorporates changes approved by the AOC WS Configuration Control Board." This release provided a section that related the ATO cycle to the OV-5. This section proved to be extremely helpful in creating our architecture products for the AOC conducting the ATO cycle. These products can be seen in Appendix B and the process used to create them will be covered in more detail in the Chapter 3.

It is worth mentioning here that the "baseline" AOC-WS is still changing. According to Norman, "the AOC built and stood-up at AUAB was declared to be the first instance of the AOC Weapon System, and its configuration was set as the baseline." [20] However, to meet the demands of the warfighter and to incorporate new systems alterations to the baseline at AUAB, modifications have occurred unabated. [20, 3] The AV-1 from the current MITRE architecture also states, "This architecture will continue to evolve. Currently, it reflects the architecture as we believe it exists currently and in the near future (FY04-FY06) to reflect systems already in development and that are expected to be fielded in that timeframe." Needless to say, one of our disclaimers to this project is that our study to create an executable architecture for the ATO cycle should be taken as purely academic. The conclusion and recommendations section will note how efforts to create executable architectures in the future can be more accurate and useful.

As stated in the DoDAF, "Most graphical architecture products (e.g., OV-2, OV-5, SV-1 and SV-4) permit the modeling or their respective architecture data elements using decomposition (i.e., several diagrams of the same product may be developed for the same architecture, where each diagram shows an increasing level of detail). In general, the level of usable detail increases as the perspective changes from that of the planner, to the owner, to the designer, to the builder." Figure 2.21 shows some of the rules of thumb for decomposition rules.

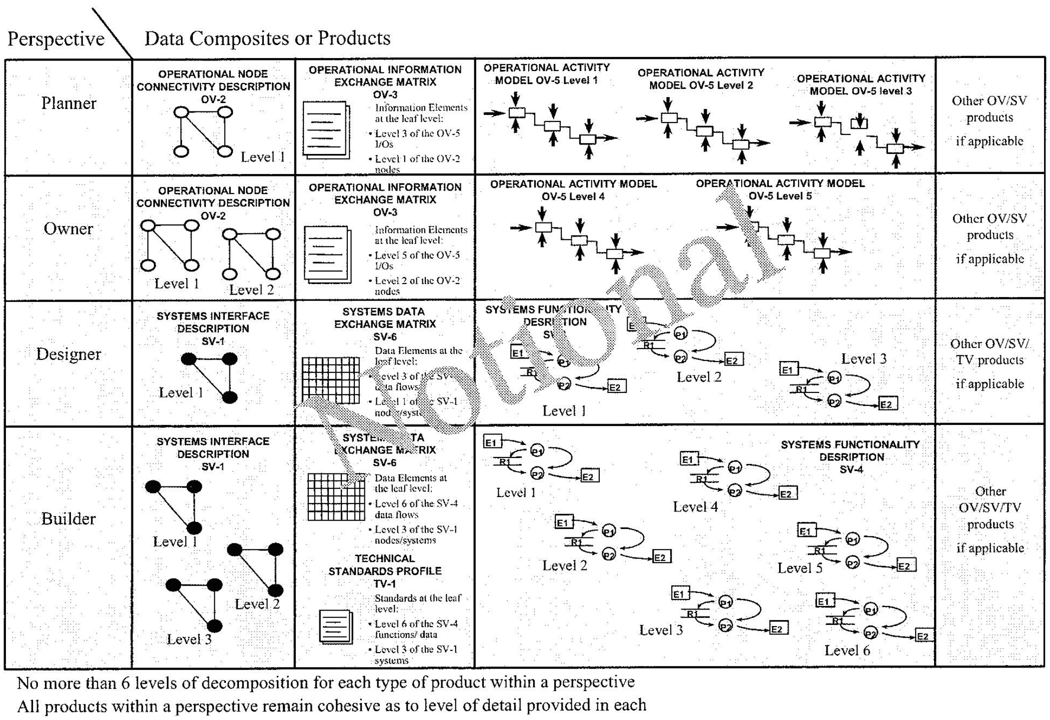


Figure 2.21 Rules of Thumb for Decomposition [7, 2-9]

In our study, the architectures we created for the ATO cycle were decomposed to the functions performed by the divisions. We decided on this level of the decomposition because the processes performed by the divisions were more eloquently represented in the MITRE architecture.

2.4 *Executable Architectures*

2.4.1 Colored Petri Nets. In an effort to quantify the distributed architecture design options, we created an executable architecture with a Colored Petri Net (CPN). Petri nets are a formal, graphical technique used to model a discrete-event system. While graphical, they can express a system mathematically as well and can be used to verify a system. Distributions can be added to perform statistical analysis on a system. Using timing, performance characteristics can be measured. Their flexibility gives them a great

deal of power as an executable architecture. Their visual nature makes them intuitive and relatively easy to understand.

Petri nets were introduced by Dr. Carl Adam Petri in his PhD dissertation at Bonn, Germany in 1962. They were quickly recognized as a powerful modeling tool, especially for modeling concurrent processes. In the 1970s, high level Petri nets, including colored Petri nets, were developed and in the late 1980s hierarchical nets were born. Somewhere in that time Petri nets had been spread throughout Europe and are used there extensively. While some American universities have become involved with Petri nets, Petri nets' influence has been minor in the U.S.

When working with CPNs, you need only remember the four basic elements to colored Petri nets: places, transitions, directed arcs and tokens. All four of these elements are present in Figure 2.22 below.

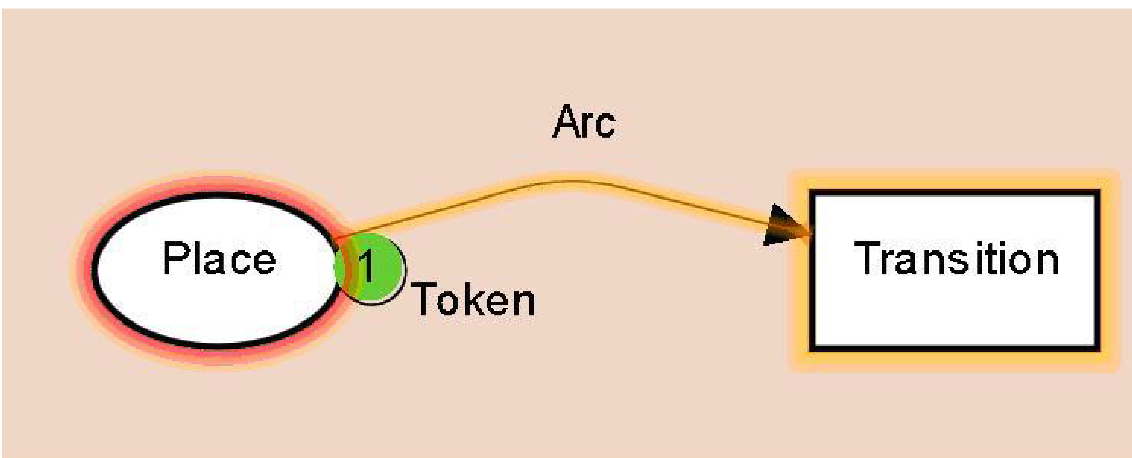


Figure 2.22 Basic Colored Petri Net

A place is always represented by a circle or ellipse. In CPN Tools (version 1.2.0), a place has three locations for inscriptions. The first is the place name which is shown inside of the ellipse. The second is the color type and is typically shown below and to the right of the place. The last inscription is the initial value which is shown above and to the right of the place. This allows the user to predefine the initial state of the CPN by defining

any tokens that the place may hold before execution. A place can model an object and can hold one or more tokens which contain data associated with the object.

These tokens can be given different “colors” or attributes that make them excellent for passing information between objects. In CPN Tools there are 4 basic color types: STRING, INT, E, and BOOL. The STRING type is a string of characters; INT is an integer (0, 1, 2, 3,); E is an enumerated type where the modeler defines the possible values; BOOL can be true or false. A place may also have a color corresponding to any color type or product of color types.

The information (token) goes from a place to a transition and from a transition to a place. A transition is modeled as a rectangle and can transmit, modify, destroy, or create new tokens. This makes token placement a non-trivial part of the model creation. A designer must decide if the token will start in a place or if it will be created in a transition. Likewise, a token can end in a place or be destroyed by a transition. Either way, a token will be in a place at some time in its existence and it will reach a transition at some time. The transition is used as the processing center for a CPN and has logic associated with it. This is where all of the work occurs.

Like places, transitions have inscriptions. The first is located in the center of the transition and is its name. Second is the guard which is contained in brackets ([]) and set the rules for when a transition will “fire” (process the tokens). The third inscription is the time delay associated with the modeled processing time of the transition. Finally, a code segment can be added to the transition. This code segment is where all of the “work” can be done to the tokens. Information can be created or used by creating or destroying tokens; randomness can be inserted; conditional logic can be used to make decisions.

The pathway followed by the tokens is an arc which can have logic that controls the flow of the tokens. This logic is in the form of conditional “if” statements that can change the contents of the tokens and even destroy tokens. The arc can have exactly one origin and exactly one destination, but tokens can travel both directions on an arc, depending on the directionality of the arc.

The model in figure 2.23 is rather simple and easy to view in a single page. Since one of the strengths of CPNs is the ability to visually inspect the model, a large model would nullify this if not for abstraction. Transitions can be replaced by sub pages that allow the details to be hidden, yet still accessible. This CPN is actually hierarchical. The "Create CPN" transition in figure 2.23 has a sub page that has the logic for this CPN. If you open this sub page, you will find figure 2.24.

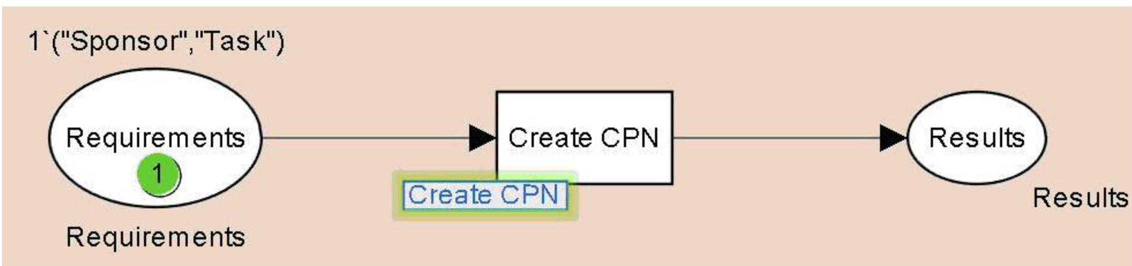


Figure 2.23 A Simple Hierarchical CPN

On this sub page all of the logic involving the process to create CPNs is visible. The first transition ("Develop Static Architecture") will take the token located in the place labeled "Requirements" and create architecture products ("OV5", "OV6a", "OV7", "OV6b") which are modeled as places. The initial token starts at the "Requirements" place and will be destroyed in the "Develop Static Architecture" transition. All other tokens are created in transitions.

This model has some useful features. If you wished to see how fast you could produce a CPN, you can add timing to the model. Timing can be used to optimize your CPN creation processes. If you are looking for useless steps in your CPN creation process, a state space analysis can identify them.

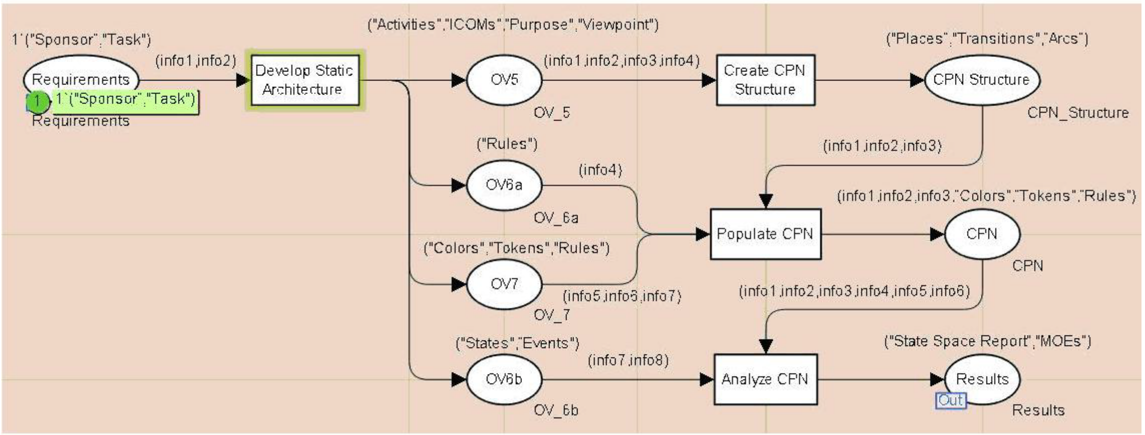


Figure 2.24 CPN Sub Page

Some things are harder to model. We found the concept of quality to be less than ideally applicable inside the model. For example, if incoming information is of questionable value, how do you model its quality and how will that quality affect a transition? This concept was important to the model we created, and continues to be an area of research scarcely documented.

2.4.2 CPN Development and Traceability. The DoDAF goes into great detail on the creation of static architectures. It even shows more than one approach by demonstrating how an object-oriented architecture can be created. It is amazingly quiet about executable architectures. Although it recognizes their importance, little content is delivered. As a result, there are many professional products available to create executable architectures, but no DoD guidance or standardization procedures to follow in using them. The DoDAF is an integrated approach and we feel it's essential to keep that continuity through the executable architecture as well.

In the construction of our executable model, we relied heavily on a paper co-authored by Dr. Levis, "C4ISR Architectures: II. A Structured Analysis Approach for Architecture Design." [19] This paper has key information on how to convert a C4ISR architecture into an executable architecture using Petri nets by mapping elements within

views to CPN model components. We will give an abbreviated version of the process shown in Figure 2.25 here. The static architecture products needed to build and analyze a CPN are the OV-5, OV-6a, OV-6b, and OV-7. The OV-6b is used in the analysis while the other three are used to construct the CPN.

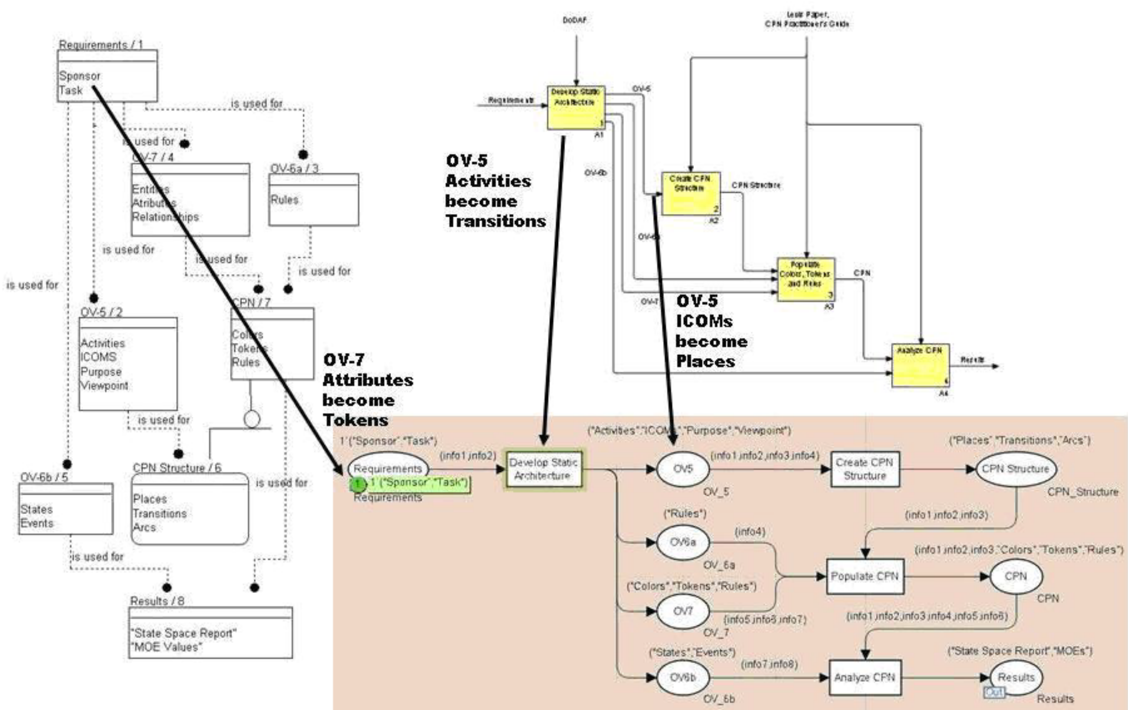


Figure 2.25 Developing an Executable Architecture from a Static Architecture

The places on the Petri model come from ICOMs off the OV-5. The activities from the OV-5 become transitions in the model. The arcs, then, are just ties between the two that follows the same flow as that of the OV-5. This part is so simple and straight forward that it could easily be automated. The hierarchy of the CPN will match that of the OV-5 where decomposed activities correspond to substitute transitions representing sub pages of the CPN. The CPN top level in Figure 2.25 corresponds with the OV-5 context diagram.

Creating tokens is where the fun begins. The tokens themselves can come straight out of the OV-7 (Logical Data Model). However, where and how they are placed is not always straight-forward. A token is assigned from the attribute of an OV-7 entity. This entity corresponds to an ICOM on the OV-5 which in turn corresponds to a place in the

model. Therefore, the token resides (at some time) in the corresponding place. If the same attribute is found in many different entities in the OV-7, you should expect to see it flow between many places on the model. Tokens that exist in only one entity on the OV-7 would be expected to be found in only that corresponding place. Deciding where a token will begin and where it will end (places vs transitions) is not as straight-forward as building the structure.

In a CPN, tokens have "colors" as well. These colors are derived from the OV-7. The OV-7 attributes have types (integer, string, etc.). These types can be put together to form the token's color. The token in our example is ("Sponsor", "Task"). These attributes from the OV-7 are both strings so the color is "STRING*STRING."

For a transition to fire, the "guard" rules must be satisfied. These guards are derived from the rules of the OV-6a. The logic on the arcs is taken from the same view. In DoDAF, these rules show how information is passed through the system, and they do the same for the CPN model.

The last architecture product is the OV-6b. This is not used in the construction of the CPN, but is used to verify the model. A state space analysis is made with CPN Tools that shows how many states can never be reached (dead states), how many states can never be left (infinite occurrence sequences), how many nodes there are, and how many transitions there are. All of this should match the OV-6b. Obviously, this means there should be no dead states or infinite occurrence sequences.

2.4.3 Arena. Colored Petri Nets are just one example of executable architectures. Since CPNs are really just a modeling tool, we identified another modeling tool that could also be used to provide insight into static architectures. Arena has strengths that give it some advantages over CPN Tools. Arena's primary strength lies in its ability to rapidly run multiple replications of a variety of model configurations in a batch mode. This type of Monte Carlo simulation analysis is extremely useful in identifying critical parameter values and other important configurations aspects.

Arena is a general purpose, visual simulation tool marketed by the Rockwell Corporation. It provides a hierarchical system of modules which represent common simulation constructs. At the top end of the hierarchy are templates of common practices which are constructed of numerous modules. At the most detailed level, user-written functions in general purpose programming languages such as Visual Basic or C/C++ are allowed. This allows for the rapid development of customizable simulation models for a variety of different analysis applications. The reader is referred to the Arena product documentation [2] for a complete description of the product. Here we offer a brief summary of the key Arena components utilized in this thesis.

An Arena model typically exists of several basic components, organized to model the process of interest to the user. The components, or blocks, are connected together to represent a flow which reflects the operation of the system of interest. *Entities* are those items which will flow through your simulation model. They may represent items of raw material, pieces of data, customers, or any other dynamic object of the simulation. [27, 24] Entities possess common *attributes* which are used to describe them, and whose values are unique to each instance of the entity within the system. If, for example, your entities are customers, then some attributes may be account numbers, contact information, or order histories. The values of these attributes may be set or modified as the simulation progresses through the use of *assign* blocks. Entities are created by *create* blocks in order to represent their entry into the system being simulated and disposed of by *dispose* blocks to represent their exit from the system.

The next common element to an Arena model is a *resource*. Resources represent those scarce items which entities require in order to progress through the model (similar to OV-5 mechanisms). Returning to our example where entities represent customers, a common resource might be tellers or other service personnel required in order to service the customers. Each entity will *seize* the number and type of resources it requires and then *release* them when it is finished. If the required number or type of resources is not available, the entity will be forced to wait until they are.

Processes are the next important element of an Arena model. A process typically represents the interaction between entities and resources. Though there are several types of processes offered by Arena, the most commonly used in the *seize-delay-release* process. This implements the procedure described above in which an entity seizes one or more resources, delays for an arbitrary amount of time, and then releases the resource(s) and exits the process. The length of the delay may be a constant, or it may be a random variable drawn from any of a number of common statistical distributions. If an entity enters a process and the appropriate number or type of resources is not available, the entity will enter a queue and wait until such time as they are.

Many systems can be simulated to a high degree of sophistication using only the elements described thus far. The Arena model used in this thesis, which will be described in detail in Chapter 3, used only a few more. Chief among these are the *hold* and *signal* blocks. A hold block does just what the name implies, it holds any entity which enters it until it either receives a signal, or detects that a certain condition exists. Signal blocks provide the impetus for hold blocks to release one or more of their entities. These blocks work in unison to control the flow of entities through the system.

Another pair of blocks which work together are the *batch* and *separate* blocks. Batch blocks combine multiple entities together into one entity which then proceeds through the system. Correspondingly, separate blocks split existing batches apart so that the individual entities may then pursue their own courses through the system. Finally, separate blocks are often followed by *decide* blocks which can direct entities down one of multiple possible paths. A decide block may make its decision based on a certain condition, such as the value of an attribute, or based on a certain probability assigned to each potential path.

While a model is running, Arena automatically gathers a variety of statistics on the behavior of the system. Average waiting times, queue lengths, and resource utilization rates are all gathered automatically. Additional statistics can be captured using *record* blocks, and data such as attribute values can be written to external files using *read/write*

blocks. In a typical Arena model run, a number of replications, or independent executions of the model will be run. Arena gathers all of these statistics across the replications, and reports average values along with % confidence intervals.

A small example may be useful in demonstrating the basic function of an Arena model. In our example system shown in Figure 2.26, students will arrive for a field trip to a museum. A limited number of buses are available to transport students. Once at the museum, a limited number of ticket agents are available to issue tickets to the students prior to their entering the museum.

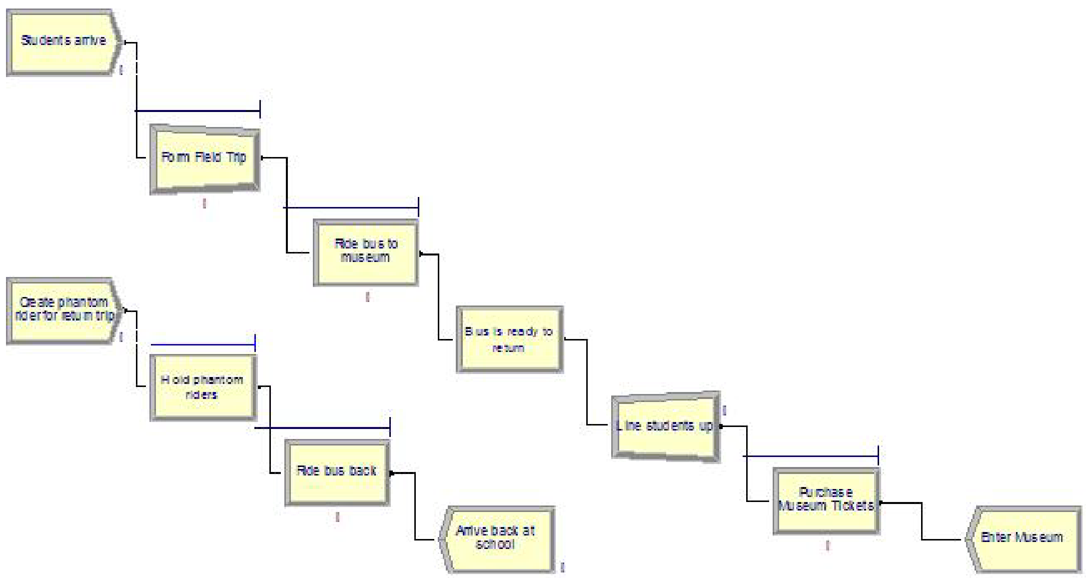


Figure 2.26 A Sample Arena Model of School Field Trips

The block in the upper left-hand corner of the figure labeled "Students arrive" is a *create* block. This create block utilizes a Poisson distribution to determine the interval between the arrival of individual students, which are one of our entity types. A second create block labeled "Create phantom riders for return trip" begins a parallel track below. This creates a large number of "phantom" riders at the very beginning of the simulation. These entities obviously are not present in the physical system, but are used to make our

simulation behave more realistically. As soon as these phantom riders are created, they go into a *hold* block, where they wait for a signal.

Returning to the top track, we find the *batch* block labeled “Form Field Trip.” It groups individual students into batches of 25. Once this number of individual students has arrived, a single entity representing the group will be allowed to proceed. This batch will then arrive at the “Ride bus to museum” *process* block. This process seizes one resource of type bus for an amount of time drawn randomly from a normal distribution which has been constructed to accurately represent the transit time to the museum.

Once this amount of time has elapsed, the batched entity will exit the process block and enter the *signal* block labeled “Bus is ready to return.” This will send a signal to the hold block (mentioned above) instructing it to release one of the phantom rider entities. This entity will then enter the “Ride bus back” process block. This will seize the bus resource which was just released when the batched students exited the “Ride bus to museum” process block, holding it for an amount of time representing the bus’s return trip. Upon exiting this block, and thus freeing the bus resource for another trip, the phantom rider entity will be disposed. Simultaneously, the entity representing the batch of 25 students will enter the “Line students up” *separate* block, which will split the batch up into its 25 constituent student entities. Each of these entities will enter the “Purchase museum tickets” process blocks, where they will seize one of five available ticket agent resources. If all five ticket agents are busy, the remaining students will queue until an agent is available. Finally, the student entities exit the system by entering the *dispose* block labeled “Enter museum”.

This simple model can be executed to predict such statistics as the utilization rate of the buses and ticket agents, or the average wait time of students. By varying the number of ticket agents, buses, or student batch size, the overall effect on these statistics can be investigated. This is such a common application of Arena models that a utility is included with the system known as the Arena Process Analyzer, or PAN. PAN allows the user to designate a range of values for variables within an Arena simulation, such as the number of

resources available or the parameters defining a delay distribution. Multiple replications can then be run at each of the design points and the corresponding statistics gathered. The appropriate variables are automatically adjusted between runs without any further intervention required from the user.

In order to run PAN, the user must first define a number of *scenarios*. A scenario consists of specifying values for the controls, such as variable or parameter settings, and determining which responses you wish to observe. Figure 2.27 below shows the PAN interface with a sample set of scenarios defined.

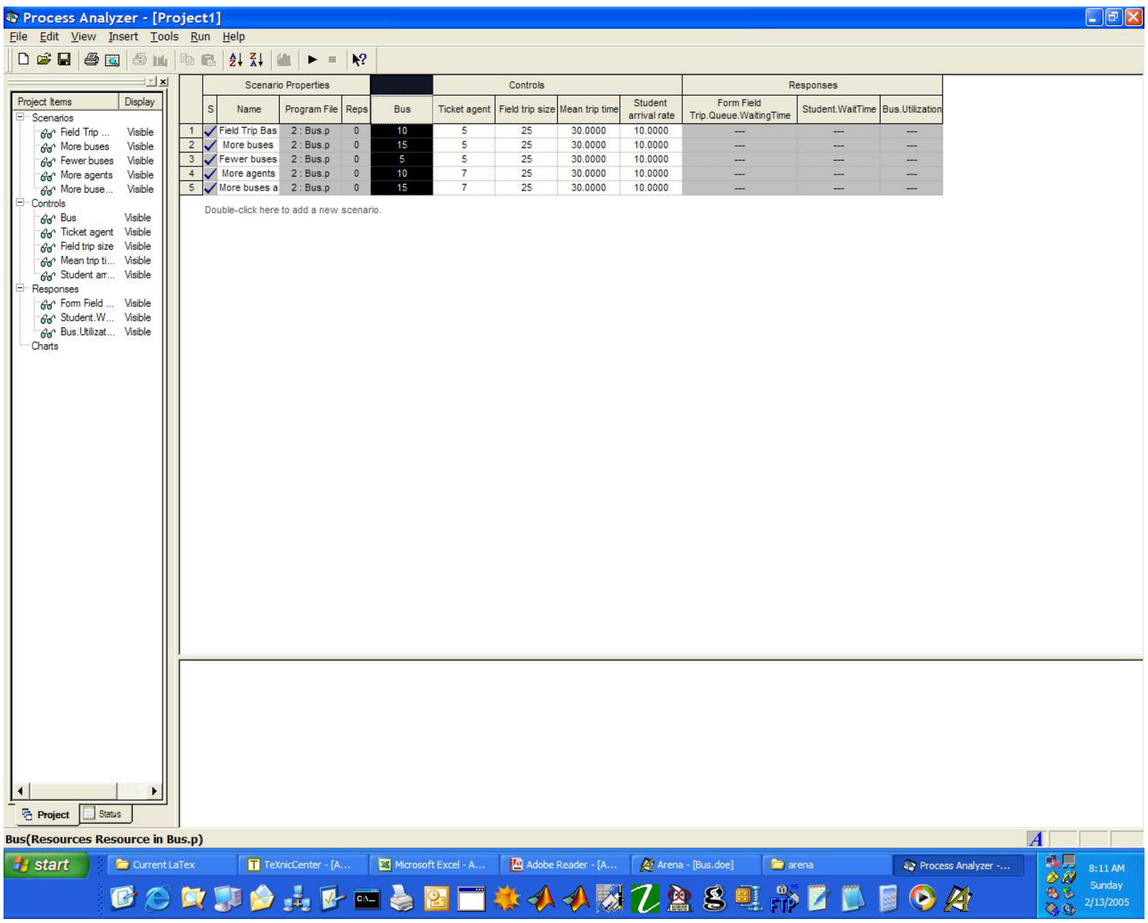


Figure 2.27 A Sample PAN Configuration for School Field Trips

In this PAN example, five scenarios (system configurations) have been defined. First among these is the baseline configuration of the system, which will be the basis for all

comparisons. The remaining four scenarios implement differing values for the number of buses and ticket agents at the museum. These are shown in the relevant columns under the Controls section of each scenario line. Further scenarios could be defined if so desired. For each scenario, data will be gathered on the waiting time to form a field trip, the amount of time students spend waiting, and the utilization rate of the school buses. These can be seen in the Responses section of the scenario definitions.

Once these scenarios have been designed, the user can simply direct PAN to run and it will execute the number of replications defined in the original model for each of the scenarios. PAN will automatically adjust the values of the relevant control and gather data on the specified responses and report these back to the user. The whole process runs in a batch mode with no user intervention required between scenarios.

III. Methodology

3.1 Scope and Assumptions

3.1.1 Scope. In order to attenuate the complexity of the Air Operation Center, and even the air tasking cycle, the process must be adequately scoped. For this research, the focus lies on the first four phases of the air tasking cycle which pertain to the development and dissemination of the ATO. Scoping to this level, there are only a few logical configurations to test. As previously mentioned, the impact of moving a complete division to another geographic location will be analyzed. The Intelligence, Surveillance, and Reconnaissance Division (ISRD) and Combat Operations Division (COD) are not going to be relocated for this study. As previously described, the ISRD has nested components in all of the other divisions. It will be assumed that the components that are nested in the Strategy Division (SD), Combat Plans Division (CPD), and Air Mobility Division (AMD) will remain nested as those divisions are relocated. The COD is solely involved in the fifth phase of the air tasking cycle, or the force execution phase. This is outside of the focus of the ATO development and dissemination. Nonetheless, it would not be logical to take the division responsible for last minute ATO revisions and monitoring joint force execution away from the fight.

The remaining divisions that will be relocated are the Strategy Division, Combat Plans Division, and Air Mobility Division. Four configurations were developed and tested. The first was to allow all divisions to be forward deployed. This created a baseline for how business is done today. The three other configurations involved relocating one or more divisions. The rationale for moving divisions for the other three configurations has been provided below.

Through discussions with Major Paul Lambertson, former Chief of C-17 Tactics in the CAOC, it would seem logical to examine relocating the AMD to the Tanker/Airlift Control Center (TACC). The relocation is validated by the available support of the TACC, since the Air Mobility Element is deployed as a liaison element of the AMC TACC. The

functionality of the TACC could be modified in order to accommodate the communication needs to support the AMD mission. This would require two communication nodes, one at the forward deployed AOC and one at the TACC.

The third configuration is based upon the footprint of the AOC as well as the phases in which the divisions are involved. This configuration is based upon moving the SD and CPD to a common reachback location. This would also only require two communication nodes, one at the forward deployed AOC and one at the reachback location. In examining the notional floor plan of the AOC as shown in Figures 3.1 and 3.2, it can be seen that the SD and CPD are self contained divisions. It would be fairly easy to move these divisions out of the floor plan without significantly effecting neighboring flows of information. Also, recalling the organizational involvement in the air tasking cycle, the SD and CPD are both involved in the early target development phase.

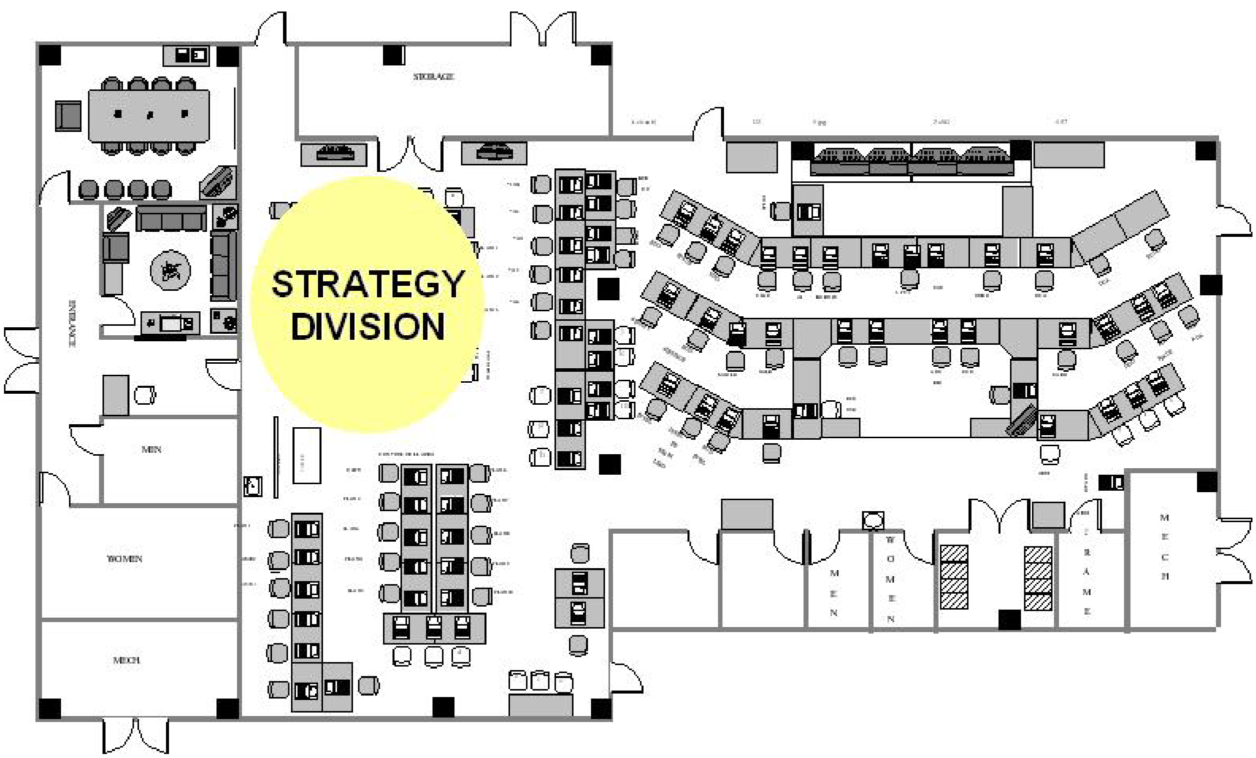


Figure 3.1 Notional Strategy Division Floor Plan

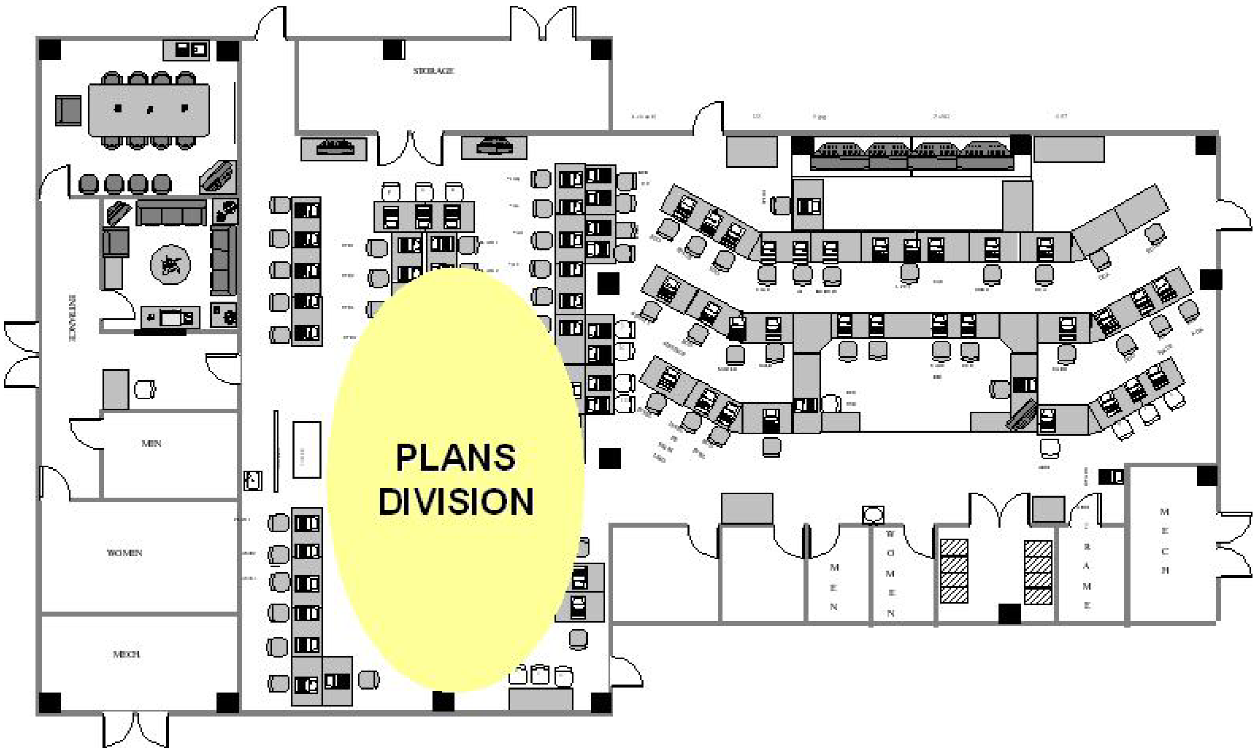


Figure 3.2   Notional Combat Plans Division Floor Plan

The last configuration combines the second and third options. This would mean that the AMD is located at the TACC and the SD and CPD are located at the same reachback location. The COD and ISRD would be forward deployed along with the JFACC/JFC and other mission essential leadership personnel. This configuration is complicated by requiring three nodes to be in constant communication.

These four communication configurations can be improved, and more configurations can be added, if the divisions are decomposed to the team level. Such an effort would increase the complexity of the communications model exponentially. The configurations we used have been summarized in Table 3.1 below.

3.1.2 Assumptions. There are a number of assumptions that were required to model the AOC in the previously described configurations. These assumptions have been enumerated below. Deviations from these assumptions will complicate the system and

Table 3.1 Configuration Summary

Cofiguration	SD	CPD	AMD
1 (Baseline)	Deployed	Deployed	Deployed
2	Deployed	Deployed	TACC
3	Reachback	Reachback	Deployed
4	Reachback	Reachback	TACC

provide basis for further research. The impact of these assumptions will be discussed throughout this chapter in the appropriate sections.

1. One ATO cycle modeled, instead of four simultaneous ATOs
2. The minimum (baseline) time to develop an ATO is the standard 72 hours
3. Delays will be from a discrete distribution
4. The number of communicating nodes will impact the range of the delay distribution
5. The preceding work of the past AFIT student group on grounding and communication mediums is a validated source for quantifiable MOEs
6. Time Sensitive Targets will not be involved in the Air Tasking Cycle
7. The Air Tasking Cycle is a perfectly linear cycle, i.e. there is no concurrent processing
8. Revisions and feedback between divisions or teams would need to be modeled at a lower level of abstraction
9. Organizational involvement will be at the division level

3.2 Static and Executable Architecture Development and Utilization Process

Many challenges were faced throughout the process of developing and employing the executable architecture. A simple four step process was outlined for developing and utilizing the executable process. This process is shown in Figure 3.3.

Each step had its own unexpected difficulties that could impact the direction of the research and any subsequent steps. The first step was to develop the appropriate

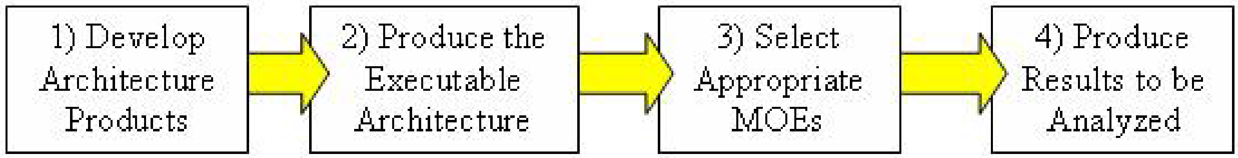


Figure 3.3 Model Development Process

architecture products required to build an executable architecture. Fortunately, there is a wealth of information available on the AOC. The prominent sources are the Air Force Operational Tactics, Techniques, and Procedures (AFOTTP) manual, the Air Ground Operation School, and MITRE (in support of ESC and AFC2ISRC). The problem with having this extensive knowledge base is trying to reconcile the various perspectives of the air tasking cycle. The AFOTTP is structured by organizations and present the process throughout the responsibilities of the division. MITRE developed a suite of DoDAF products to represent the complete AOC. The latest version release (increment 10.1) of the architecture included the mapping between the complete AOC OV-5 and the Air Tasking Cycle. This provided the necessary foundation to develop the three remaining products.

The second step involves transforming the OV-5, OV-6a, and OV-7 into an executable architecture. Recall that the OV-6b is used for verification of the CPN's functionality. The process of transforming a static architecture is not a novel idea since the methodology has been explicitly laid out by Dr. Levis as seen in Section 2.4.2. . However, there is one colossal void in using this methodology for our intended purpose. The ability to assess different AOC organizational configurations has not yet been attempted. This required a new concept of introducing mechanisms from the OV-5 as resources in the CPN. Mechanisms are not passed between activities in an OV-5. Similarly, they should only be located in a single place in a CPN. The resource, or division in our situation, will be required to fire a transition and then will be passed to its originating place. The resource can have attributes that impact the processing that is done within the transition. We will elaborate this concept later.

The third step is the appropriate selection and implementation of MOEs. The CPN that has been developed only contains the operational aspects of the air tasking cycle. A MOE that would be influenced by split AOC operations needed to be carefully selected and implemented. Originally, ATO quality was desired to be the MOE. We quickly discovered that the area of quantifying quality is an ill-defined MOE. This is an active area of research that was deemed outside of the intended goal of the thesis. Next, implementing a MOE from a precursor Systems Engineering research group was pursued. This previous group developed a matrix that described the capabilities of various communication methods and the corresponding communication requirements of various AOC processes. This "grounding" concept was discussed in Section 2.2. In order to use the communication requirements matrix, each responsibility needed to be mapped back to one or more transitions. The mapping would allow rules to be developed to determine which transition did not meet communication requirements, and by how much. Such a strategy would work if a more robust code segment was developed for CPN Tools. Currently, only one "IF, THEN, ELSE" logical structure is permitted. The implementation of the rules within CPN Tools to track multiple MOEs is beyond the scope of our research. Instead, the same requirements were preprocessed to be used as an input to CPN Tools. Each configuration has a different number of missed requirements and could be assessed a time delay in order to compensate. This resulted with time being the final MOE. The details of incorporating time as the MOE will be discussed in Section 3.4.1.

3.3 AOC Architecture Products

As previously discussed, DoDAF Vol I defines the minimum set of products for an integrated architecture as: AV-1, AV-2, OV-2, OV-3, OV-5, SV-1, and TV-1. Following the methodology explained is Sec 2.4.2, the architecture products required to develop an executable architecture are the OV-5, OV-6a, OV-6b, and OV-7. These will be the focus of this section due to their direct impact on the executable architecture. While the remaining

products in the minimum set are extremely important, they have little relevance on how to develop an executable architecture.

The AOC Architecture is controlled by the MITRE Corporation. In this thesis, our scope is the ATO production cycle, a subset of the operations of the AOC as a whole. The best source for the ATO cycle is the AOC schoolhouse at Hurlburt Field. Unfortunately, while the schoolhouse has drawn out and explained the ATO cycle in some detail, they have not developed an architecture to go along with it. Fortunately, the new MITRE architecture release maps their architecture to the schoolhouse ATO cycle. This allowed us to use the MITRE architecture, thereby transferring the responsibility of verification and validation to the experts. Therefore, our model is built from an architecture that the Air Force accepts as the most accurate.

The last issue was to reduce the MITRE architecture to fit our scope. Using the mapping provided by MITRE, we were able to create an ATO cycle architecture that agrees fully with the official AOC architecture.

3.3.1 Operational Activity Diagram (OV-5). The MITRE architecture OV-5 included more than we needed or wanted. As shown in Figure 3.4, the A0 level has 7 activity blocks of which we used just 5 and we only decomposed 3 of those. The 5 activities corresponded to the ATO cycle, but two of them had nothing to do with actually creating the ATO. These were the Execution and Evaluation activities. The third of the five MITRE activities (Task Available Capabilities/Forces) was split into two activities in our architecture (Weaponeering and ATO production) to correspond with the Hurlburt schoolhouse model of the ATO cycle.

The mapping tool that was included in the 10.1 release of the MITRE Architecture is shown in Figure 3.5. The beauty of this tool is that it maps directly into the air tasking cycle that is the focus of our model. The left side of the picture shows the air tasking cycle. When one of the six steps are selected, the corresponding MITRE activities appear on the right side of the figure. This tool was invaluable in creating our OV-5.

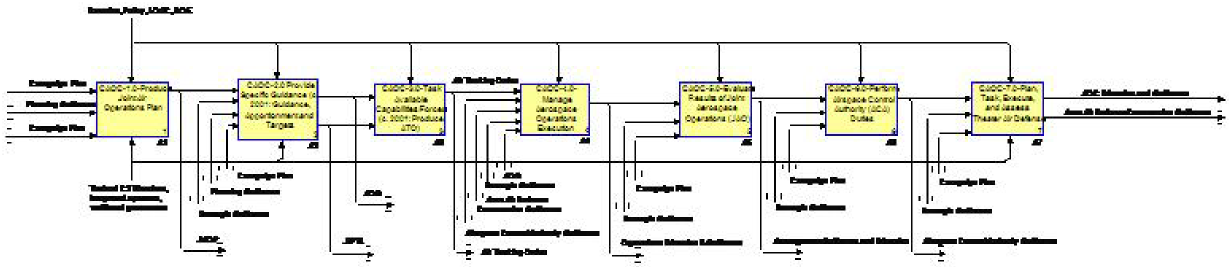


Figure 3.4 MITRE OV-5 Diagram

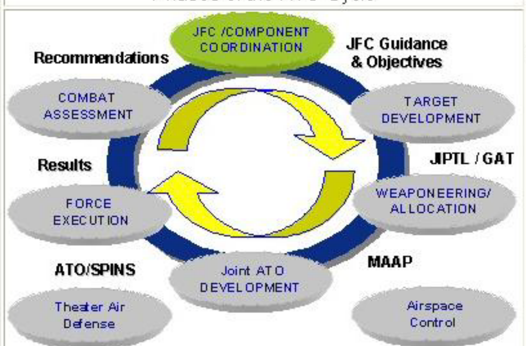
Phases of the ATO Cycle		Corresponding Operational Activities
	Clicking a phase on the left shows related operational activities on the right. Clicking an activity on the right focuses the architecture product, below, on that activity.	▪ CAOC-2.0-Provide Specific Guidance (c. 2001: Guidance, Apportionment and Ta (Operational Activity) • CAOC-2.1-Develop and promulgate daily guidance (Operational Activity) • CAOC-2.2-Develop Targets into a Joint Integrated Prioritized Target List (Operational Activity) • CAOC-2.3-Develop apportionment recommendation (Note: Decomposition assumes (Operational Activity) • CAOC-2.4.1-Submit apportionment recommendation and prioritized CTL;TNL for (Operational Activity) • CAOC-2.5-Disseminate approved product as the Joint Integrated Prioritized T (Operational Activity)

Figure 3.5 ATO Cycle

The OV-5, after realigning it for our scope, is shown in Figure 3.6. Note that even though all six steps are included in this OV-5, the emphasis remains on the first four steps that lead to the production of the ATO. The remaining two steps of force execution and assessment complete the cycle and would be potential areas for further research.

The MITRE architecture agrees with our ATO architecture at each level of abstraction until it reaches the limit of our scope with the exception of A3.1, Develop Weaponeering Solutions for Targets. Figure 3.7 shows MITRE's A3.1 which has twelve activities at this level, which violates the suggestion to limit the number of activities to nine in order to maintain readability.

MITREs A3.1 was reduced to what is shown in Figure 3.8. This was done because the first five activities had an identical output. In thinking ahead to the executable model,

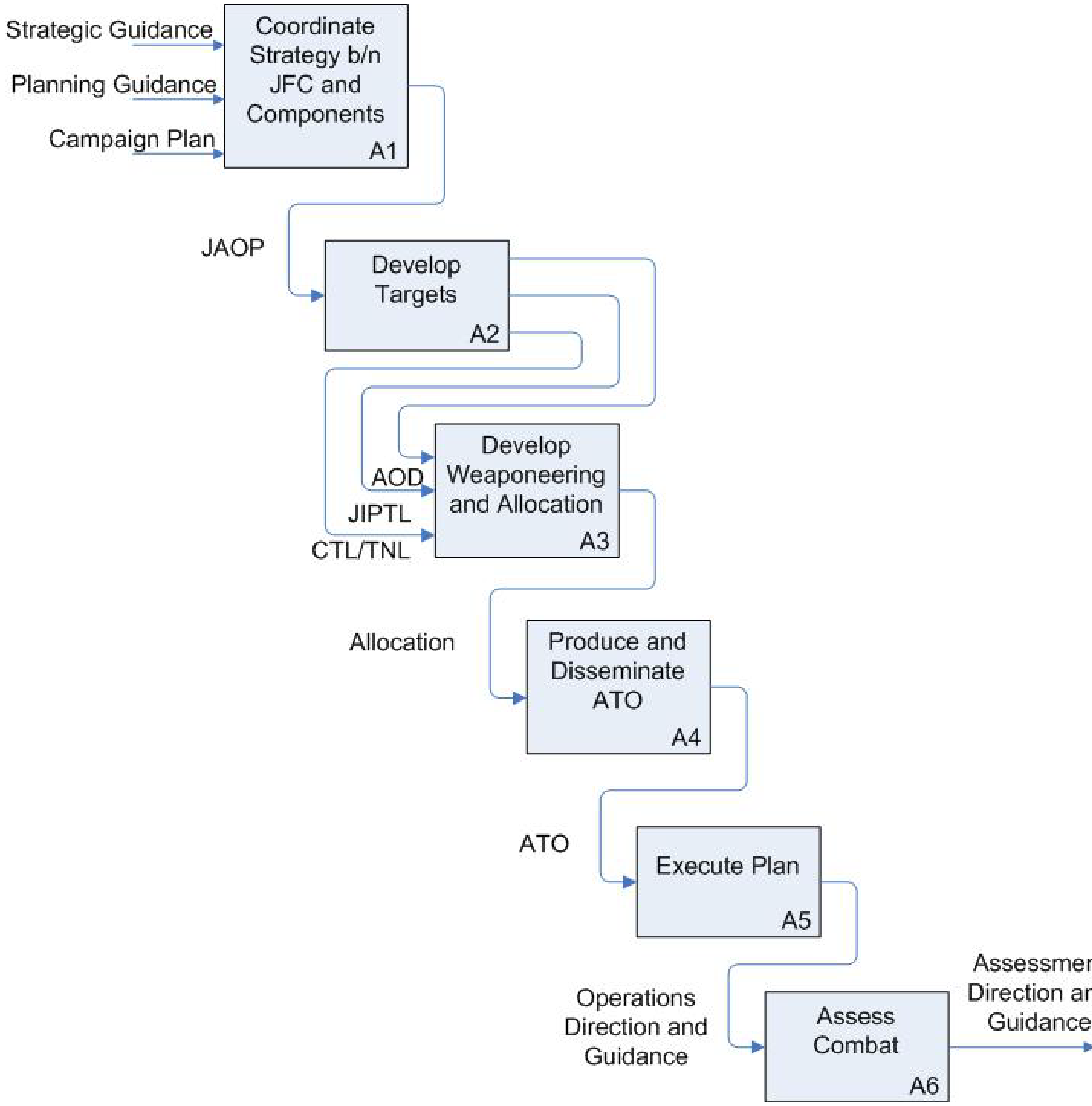


Figure 3.6 Reduced MITRE IDEF0

this would be equivalent to passing a token through multiple transitions without changing any attributes. If the output can not be uniquely identified, the activity should be omitted. We also combined the two activities that created target worksheets and target folders as "Develop Target Solutions." These modifications increased model coherency and readability.

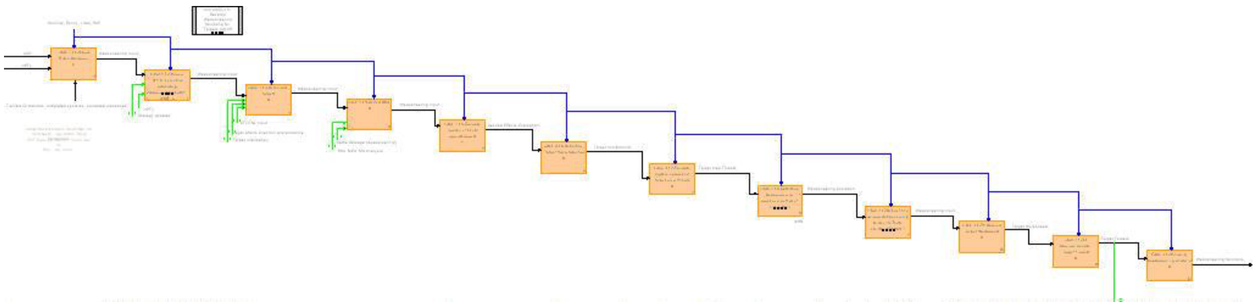


Figure 3.7 MITRE's Develop Weaponeering Solutions for Targets Decompositions

3.3.2 Rules Model (OV-6a). The Rules Model contains the necessary information to ensure that the CPN performs correctly while evaluating any MOEs. Rules should be written at the appropriate level based on their intended use. For example, rules for a mission level may consist of doctrine, guidance, and rules of engagement. Rules are written as Structured English statements and Rules are generally implemented within the transition in a CPN. Specifically, a rule can be implemented in the inscription on a transition for a guard, time delay, or code segment. It is also possible to use arc inscriptions to implement rules. For our MOE of time, all rules were implemented in the time delay portion of transitions. Generally, the Structured English rules for timing our scenario would follow the following form

IF configuration = 1 THEN delay = Missed Requirements * discrete(1,5)

The rule set for timing has been simplified to tabular form and shown in Table 3.2.

Table 3.2 Time Delay Rules

Configuration	SD Location	CPD Location	AMD Location	Delay
1	Deployed	Deployed	Deployed	Missed Req'ts * discrete(1,5)
2	Deployed	Deployed	TACC	Missed Req'ts * discrete(1,15)
3	Reachback	Reachback	Deployed	Missed Req'ts * discrete(1,15)
4	Reachback	Reachback	TACC	Missed Req'ts * discrete(1,30)

Each transition requires the appropriate divisions (per Section 2.1.3) to be available before firing. The transition then determines which configuration is being tested based on the attributes of the divisions. The transition finally assesses the delay based on the

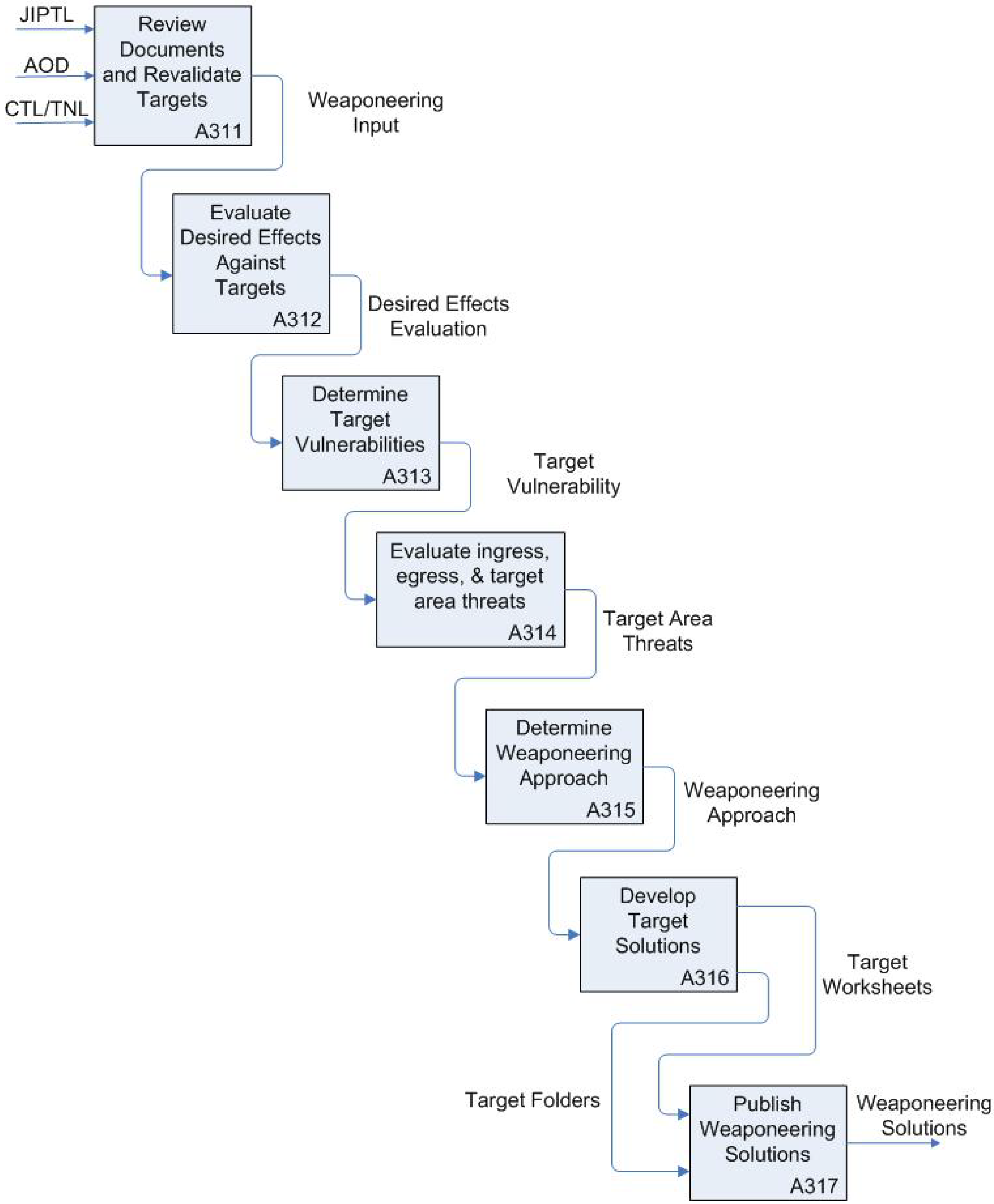


Figure 3.8 Simplified MITRE Decomposition

number of missed requirements and configuration. Figure 3.9 shows the implementation of the time delay rule for A11, Analyze Mission.

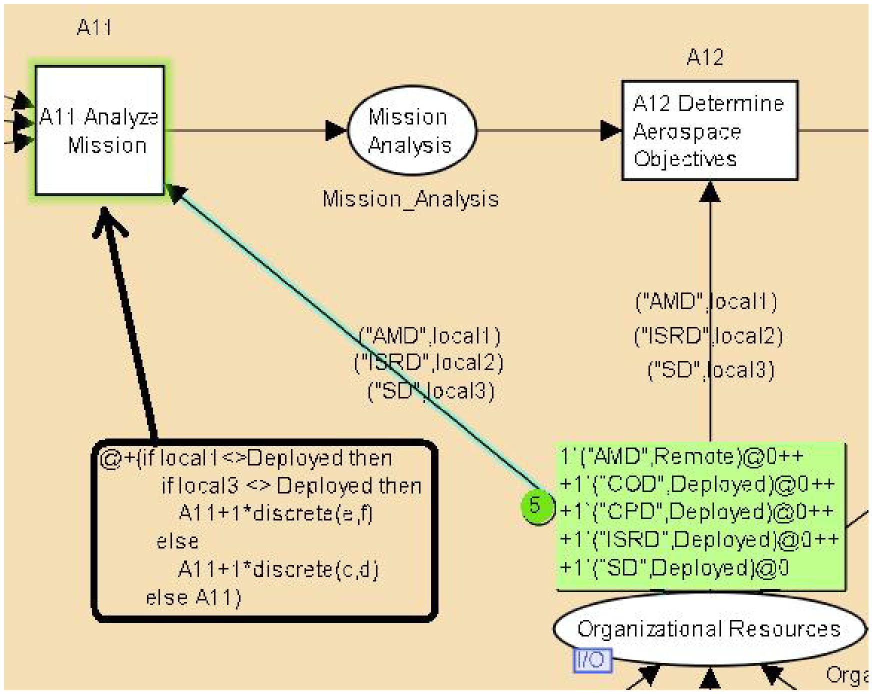


Figure 3.9 Example Implementation of a Rule

In order to fully understand how the timing rules were implemented, the Analyze Mission transition will be examined. In Figure 3.9, it can be seen that three divisions are required to activate the transition based on the three arc inscriptions between the transition and the Organizational Resources place, located in the bottom right of the figure. There are five available tokens in the Organizational Resources place, one for each division. The attributes of each token include the division name and location. All divisions are forward deployed except for the AMD, so this picture is taken from configuration two. The time delay logic is shown in the lower left hand corner of Figure 3.9. The delay logic uses nested if statements to determine the configuration and then assign the appropriate delay. It should be noted that each rule must be specifically coded for each transition, which makes implementing conditional MOEs very time consuming for large CPNs.

Additional rules were developed to ensure the flow of the CPN was correct, but were not necessary to implement in the CPN because of the linear process. It is possible for guard functions to be needed to ensure the proper flow in nonlinear CPNs. All of the rules, including rules governing process flow, have been included in Appendix II.

3.3.3 State Transition Diagram (OV-6b). The state transition diagram is used to verify that the CPN passes through all expected states. State transition diagrams can be developed at various levels of abstraction, and we modeled them corresponding to the A0 level decomposition of the OV-5. The State Transition Diagram for the Air Tasking Cycle is shown in Figure 3.10. This is verified as the CPN passes through the states in the proper sequential order. The state transition diagrams for nonlinear systems would be more difficult to use in verifying the functionality of a CPN.

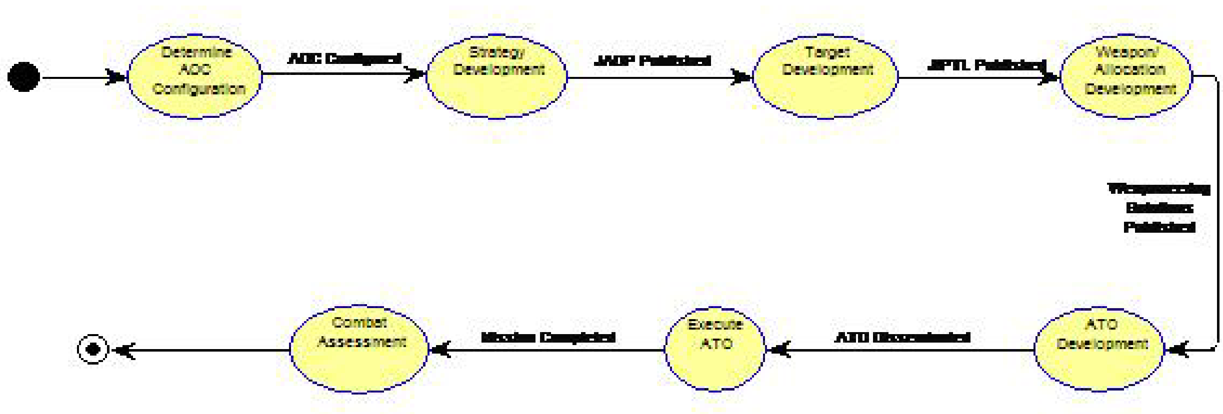


Figure 3.10 Air Tasking Cycle State Transition Diagram

3.3.4 Logical Data Model (OV-7). The OV-7 was developed in the course of our research. We see the creation of a validated OV-7 as the key remaining task in completing the MITRE architecture.

Fortunately, most of the data we needed was obtained from the data dictionary. There are many comments for the existing products that spell out the generics of the system information. For example, the OV-5 ICOM between A1.1 and A1.2 is “Mission Analysis.” The data dictionary description for this ICOM is “An analysis of: Theater

Commander/JFC mission and intent; NCA/Theater Commander/JFC determined end state; existing plans; applicable joint, multinational and Service doctrine; treaties, policies, legal restrictions, and ROE; JFACC guidance or direction; and command relationships." From this we were able to create the OV-7 entity "Mission Analysis" and fill in its attributes as shown in figure 3.11. Since our purpose was to create the CPN model, only attributes that we decided would be modeled needed to be included in the OV-7. If multiple attributes always followed the same path in the CPN, we could combine them to simplify the model and add readability. These attributes are modeled as strings instead of data because that is how they are passed in the model.

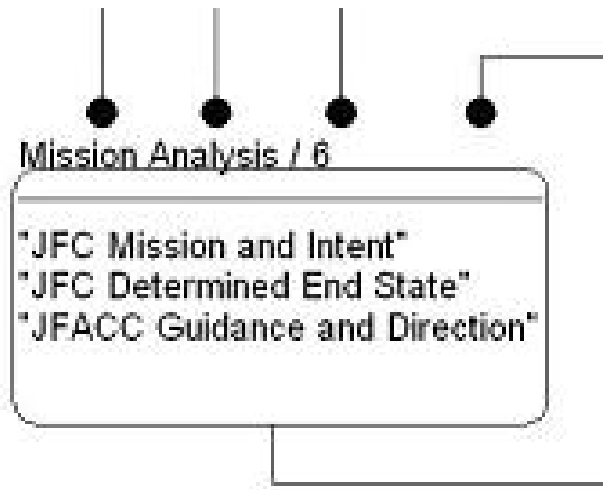


Figure 3.11 OV-7 Developed for Colored Petri Net

3.4 Executable Architecture Development

CPN Tools was selected as the CPN software of choice due to its legacy of being the most widely used CPN software program. The preceding version of CPN Tools was called Design/CPN and was only to be run in a Linux/Unix environment. CPN Tools was developed to be more user friendly and compatible with Windows. With this major revision, the majority of the built in analytic features were lost. For example, Design/CPN had integrated tools for doing statistical analysis and creating graphical output. CPN Tools has the ability to generate simulation reports and state space reports. A simulation

report contains all of the bindings of variables and the corresponding time stamp. This information is sufficient if care was taken in selecting and incorporating MOEs.

There were two reasons for adding a second executable architecture. The most obvious reason is to compensate for some analytical shortcomings of CPN Tools. Arena has the ability to generate a simulation report with many statistical results included automatically. Another key feature of is the utilization of a tool called Process Analyzer. This allows multiple configurations to be predefined and run for a set number of replications. The results then are displayed for all configurations for easy comparisons. This capability will be further explained in Section 3.5. The second reason an executable architecture was pursued in Arena is to enable a tool comparison. CPNs are the only model that has been suggested for use in developing models from DoDAF products. Although CPNs provide a crisp and clean implementation from DoDAF sources, the suitability of another model was desired for comparison.

3.4.1 CPN Development. The same process that was discussed in Section 2.4.2 was used to develop the CPN. The only additional concept is the place for organizational resources. The involvement of the five divisions has been shown on the OV-5 as mechanisms. The five divisions were implemented as tokens and located in a single place for organizational resources. Each token corresponding to a division had an attribute for the division name and the location. Location was an enumerated color set that could be either deployed, reachback, or remote. The difference between remote and reachback is that two reachback divisions are considered collocated. This would not be true if two divisions were remote, in which they would be at two geographically separated locations. This nuance is intended to allow more configurations to be developed and implemented in the future. The CPN will be presented in a top down manner, similar to the decomposition of the Operational Activity Model beginning at the A-0 level. The top level of the CPN is shown in Figure 3.12.

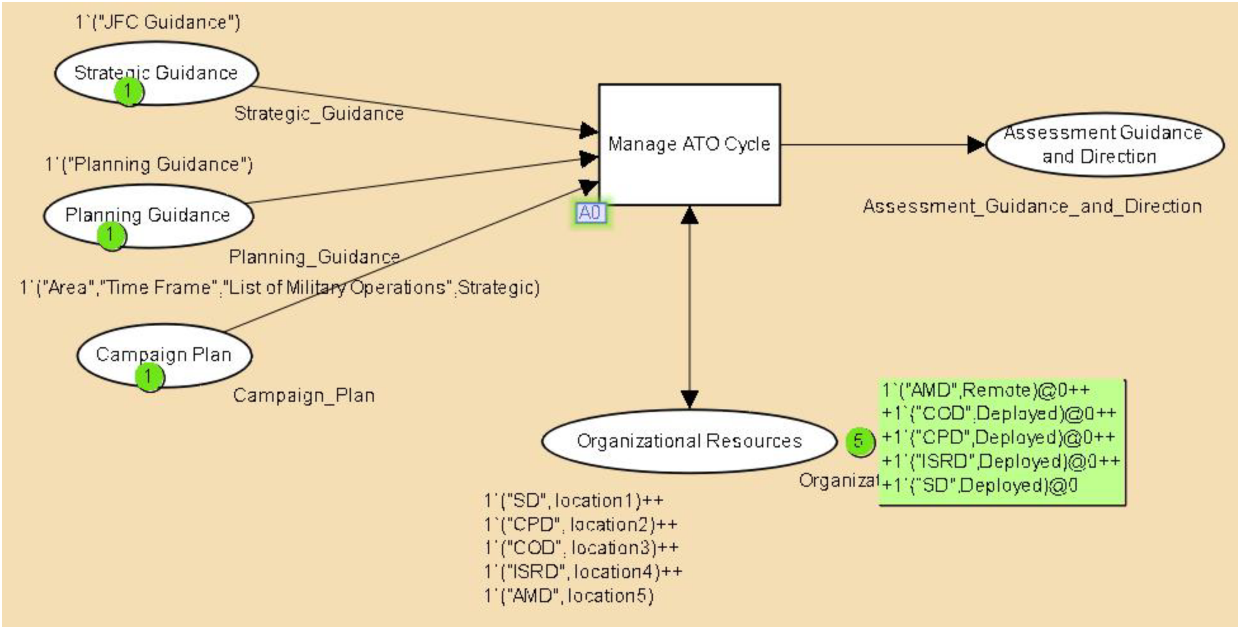


Figure 3.12 Top Level View of the CPN

The elegant one-to-one mapping of the structure of the CPN to the A-0 view of the OV-5 can easily be seen on this high level view. IDEF0 inputs become places feeding into the transitions and the IDEF0 output become places leaving the transition. The IDEF0 activity becomes the transition. This mapping provides an unambiguous method for developing an executable architecture. Figure 3.13 shows the air tasking cycle at the first level of decomposition.

There are six transitions on the A0 page. These transitions correspond directly to the six step process of the air tasking cycle as described in chapter two as well as the A1 level of decomposition of the OV-5. It is important to be aware of what transition or transitions are "primed". These are the transitions that have the necessary tokens in all of incoming places. CPN Tools highlights the transitions that are primed. If the transition is decomposed and a transition on the decomposed page is primed, CPN Tools highlights the tag on the parent transition. This can be seen in Figure 3.13 by the highlighting around the A1 tag at the bottom left of the first transition. This transition has been decomposed in

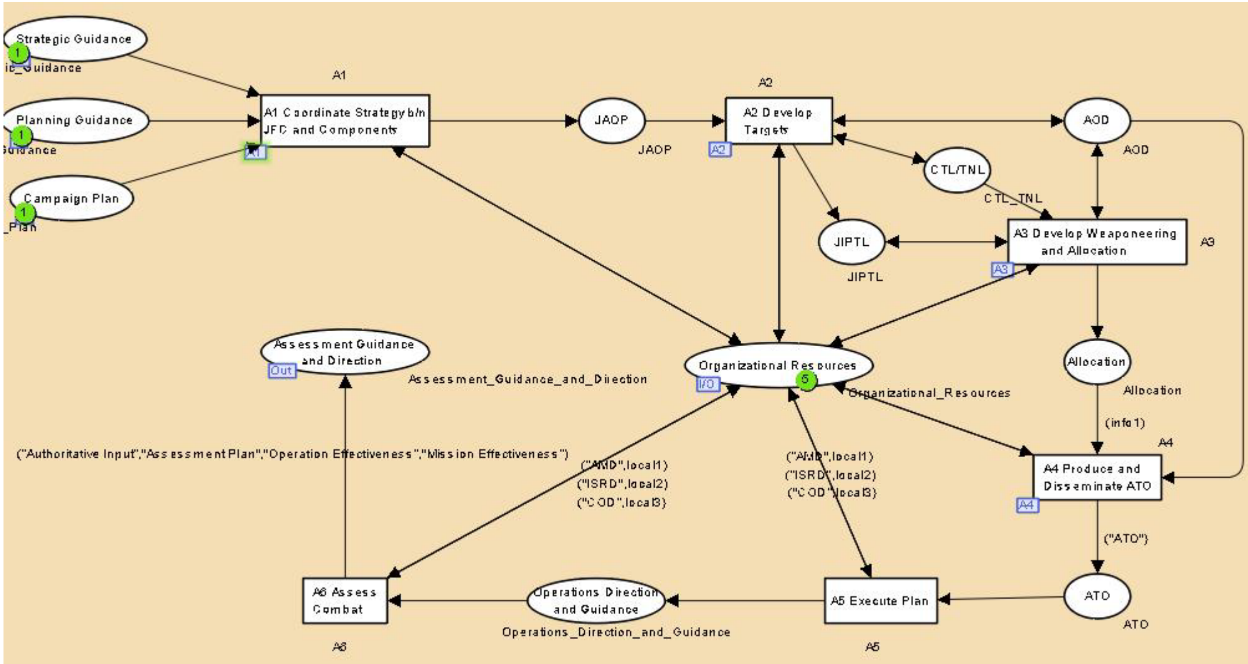


Figure 3.13 A0 View of the CPN

Figure 3.14. From this level decomposition it can be seen that the entire A11 transition is highlighted.

This is the lowest level of decomposition. This level was selected because it allows the air tasking cycle to be adequately represented without implying any restriction on how the operational activities are performed. The A2 transition is decomposed in Figure 3.15.

There are two methods of developing a CPN using CPN Tools. This is by using either a top-down or bottom-up approach. There are advantages of using a top-down approach. If a transition is selected to be decomposed, CPN Tools will place the adjacent places on the decomposition page. This helps to ensure coherency throughout the decomposition. CPN Tools also assigns the appropriate port type on both pages. For example, the Organizational Resources place requires tokens to pass in both directions. This requires that the port on the organizational resources place on the decomposed to be assigned as Input/Output capable. By following the top down approach, these nuances are automatically managed. The A3 transition is decomposed in Figure 3.16.

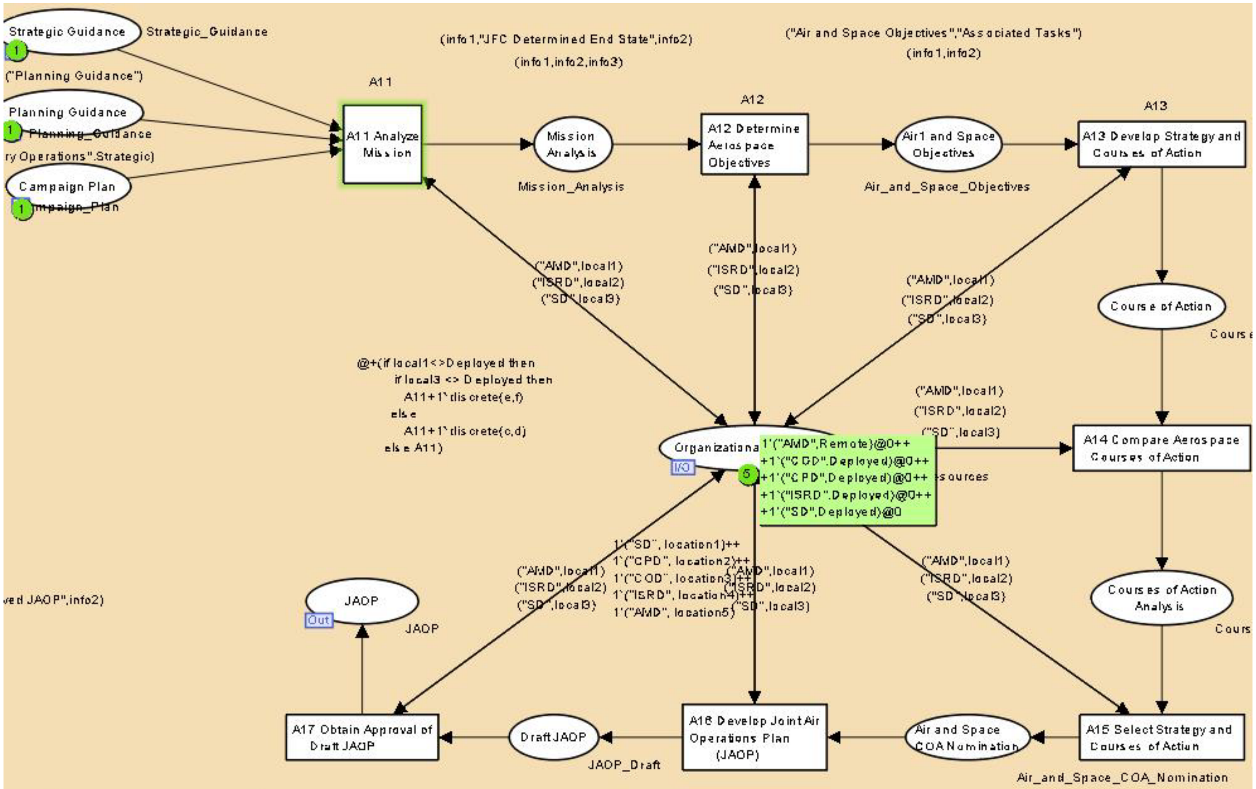


Figure 3.14 A1 View of the CPN

As transitions are decomposed it is important to keep track of what each page represents. The page numbers within the CPN model have been labeled with the same syntax that is used for activity decompositions in an OV-5. For example the page shown in Figure 3.16 would be labeled A3. The A31 and A32 transitions have been decomposed in Figures 3.17 and 3.18.

On all of the arcs between the organizational resources and any adjacent transition there appears to be only one arc. This is because all of the arcs are stacked. Then inscriptions for each arc can be seen to verify the number of arcs. In general, only one token can be passed on an arc when a transition fires. This is because the syntax in CPN Tools requires unique variables to be assigned for each token attribute. This enables all of the attributes of any token to be used in implementing rules within a transition. The last transition, A4, is decomposed in Figure 3.19.

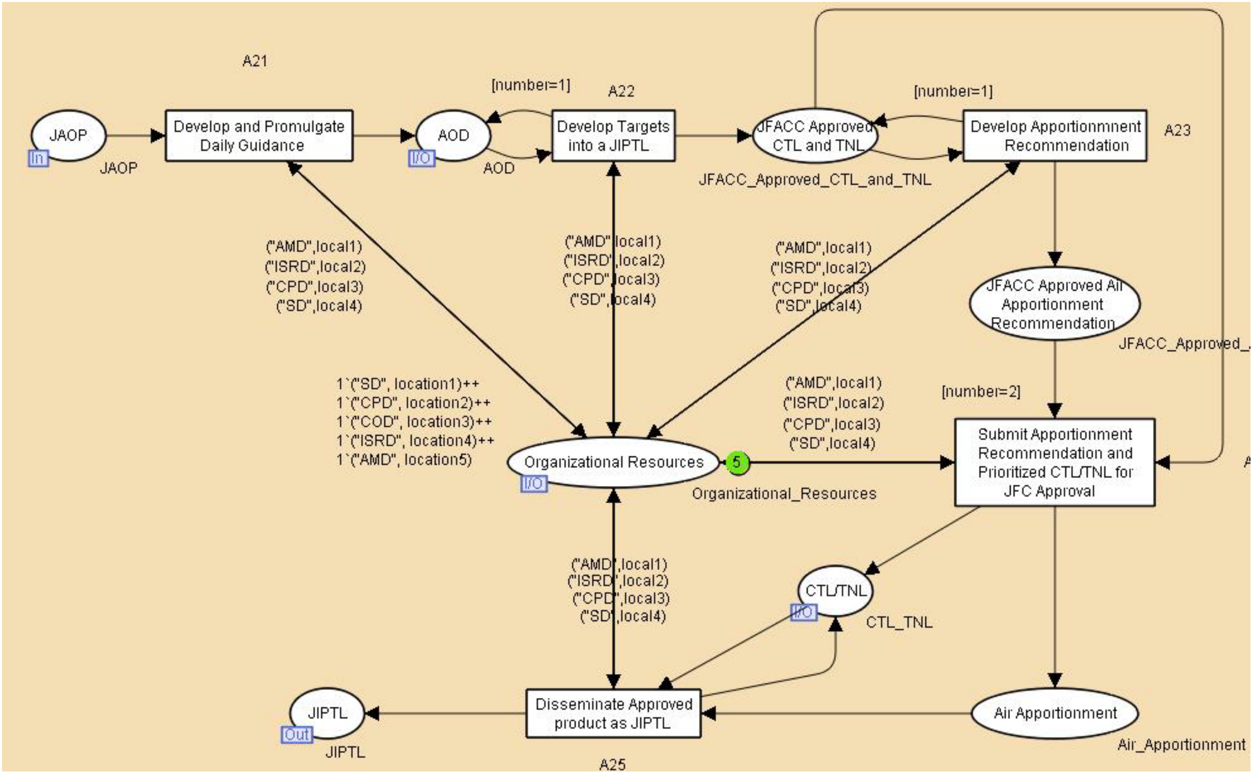


Figure 3.15 A2 View of the CPN

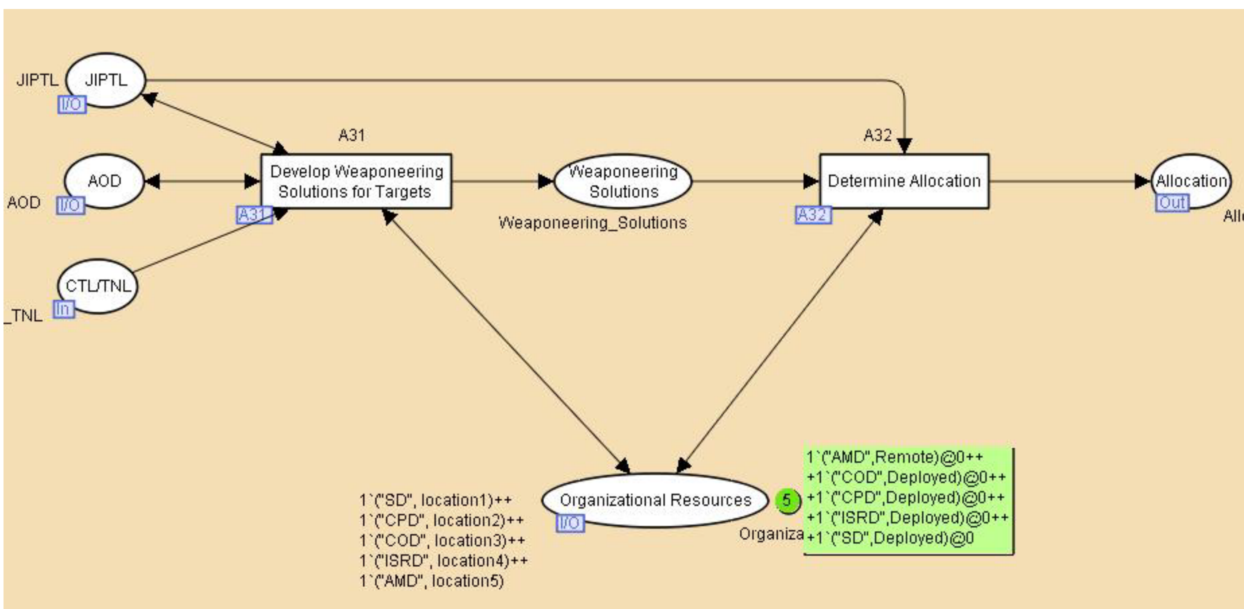


Figure 3.16 A3 View of the CPN

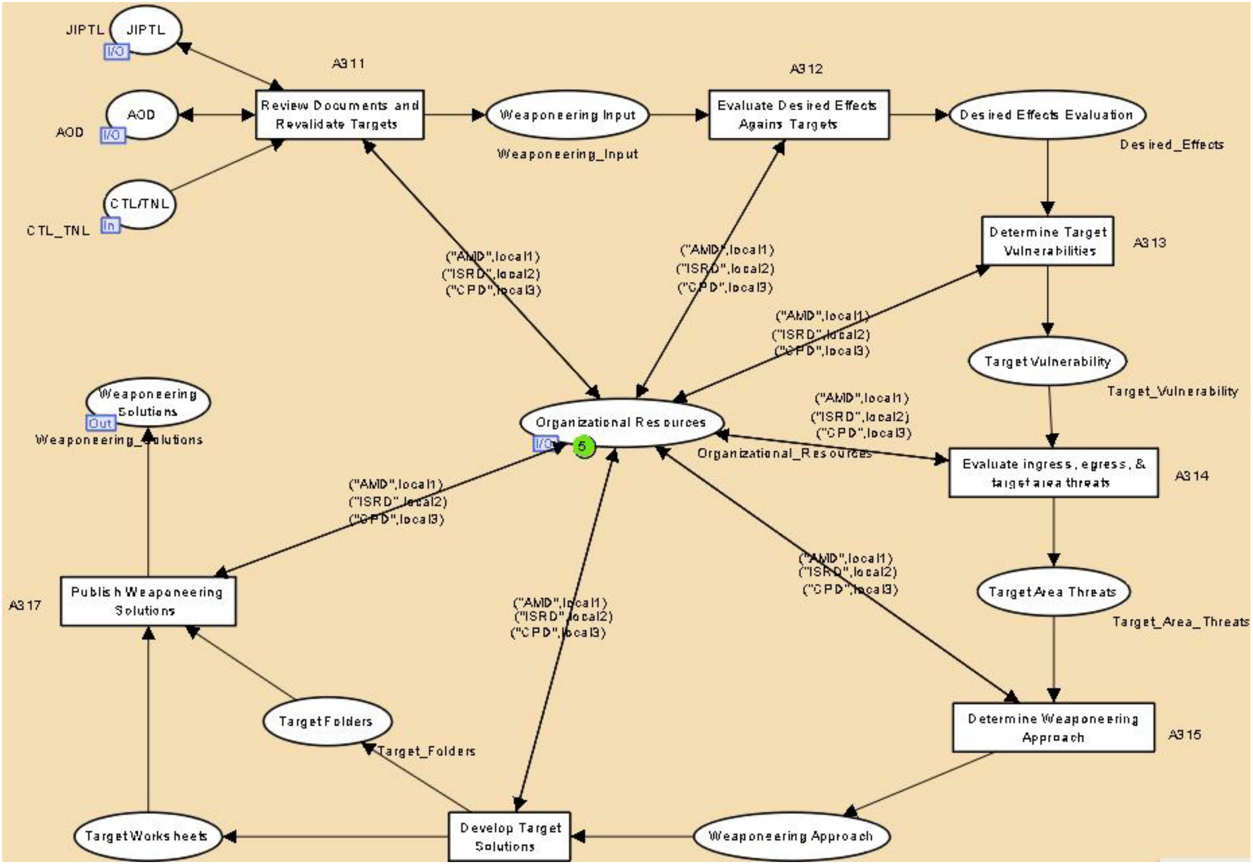


Figure 3.17 A31 View of the CPN

3.4.2 Arena Development. After completing the development of the Colored Petri Net representing the ATO production thread, we found CPN Tool's simulation capabilities to be somewhat lacking. In an effort to conduct a more thorough simulation analysis, a parallel model was developed in Arena, whose purpose was to mirror the execution of the Petri Net. This was a relatively simple modeling exercise given that the Petri Net provided the blueprint for the Arena model.

We began by defining the entities for the Arena implementation of the Petri Net. Each of the tokens from the Petri Net was mapped to an entity type within Arena, with the exception of the five tokens representing the five divisions involved in ATO production. Those assets were modeled as resources in the Arena model, as they are the scarce resources that perform activities involved in the production of the ATO. This modeling

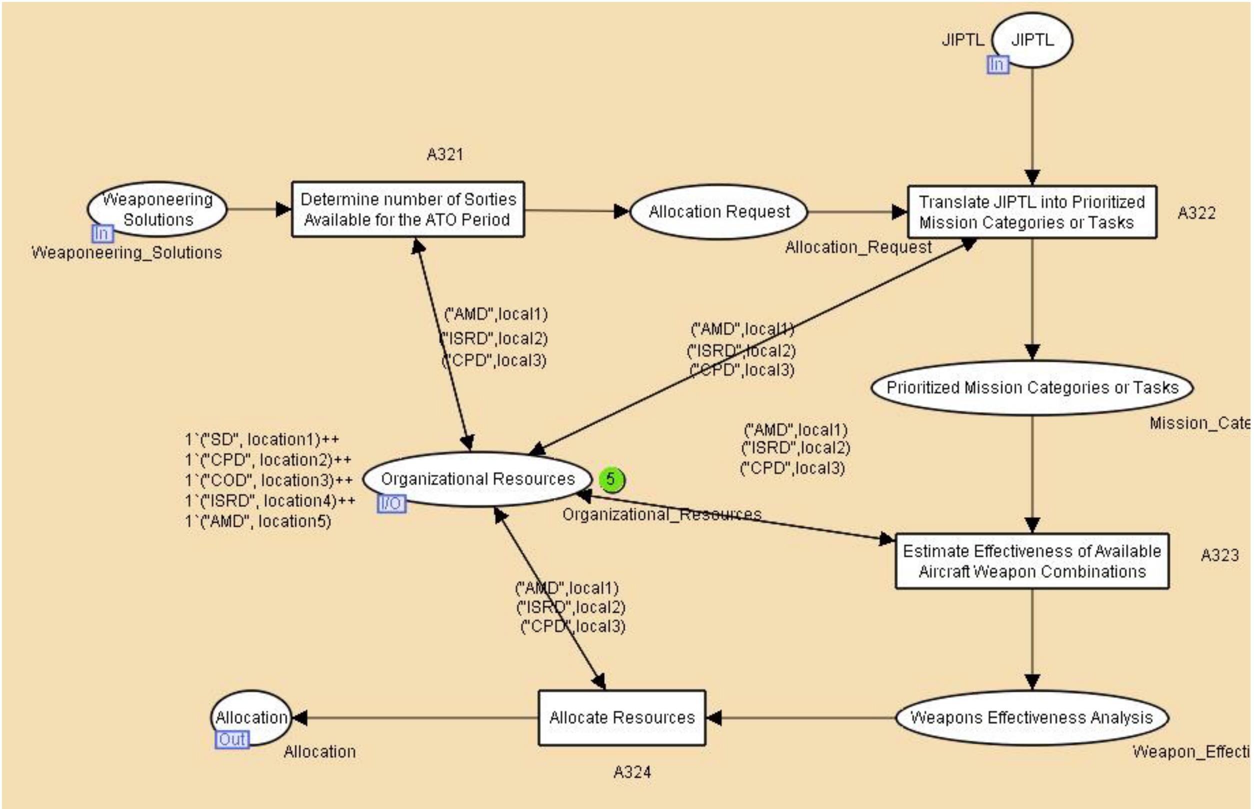


Figure 3.18 A32 View of the CPN

approach resulted in a larger number of entities than is usually present in an Arena model. Their handling was fairly simple however, as we utilized hold modules to hold the entities and remove modules to pull them from the queue at the appropriate time.

Next, the transitions associated with the Petri Net were modeled as Arena process blocks. This allowed us to associate the seizure of the resources representing the divisions at the appropriate time, and to hold them for an amount of time equal to the time delay associated with the Petri Net transition. Places in the Petri Net were not directly modeled in the Arena model. In the instances where a place had multiple transitions, entities (tokens) were batched and split, and decision blocks were used to direct entities along each of the paths representing the transitions.

Finally, a second flow was modeled in much the same manner as the "Phantom Bus Rider" construct in our earlier Arena example. This flow created a "Day" entity every 24

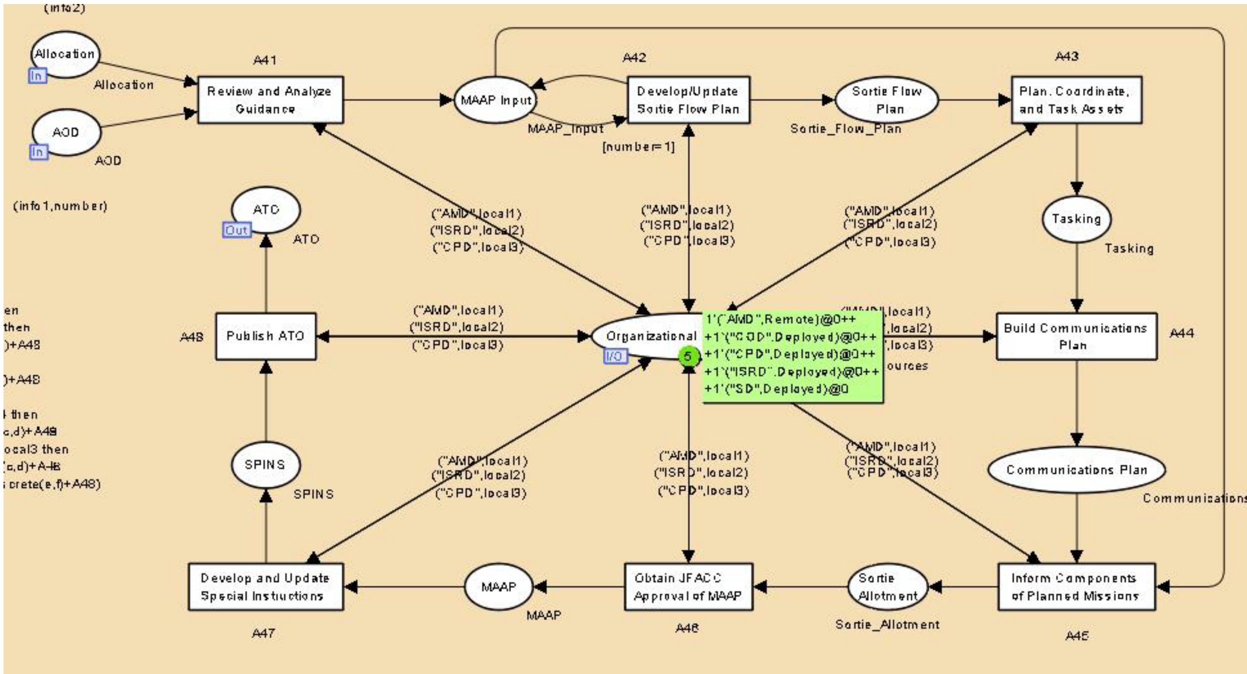


Figure 3.19 A4 View of the CPN

hours. When this entity was created, it signaled for the release of the primary guidance documents which start the ATO production cycle. In addition, each day entity also causes value of a global variable which tracks the current day to be incremented. This value can then be assigned to each token entity as a "day created" attribute. The various batch blocks within the model can then be directed to batch only entities that were created on the same day. Although simulation runs for this thesis modeled only a single execution of the ATO cycle, this construct should allow for multiple cycles to be run simultaneously.

The final Arena model consisted of 329 blocks. A model of this size cannot easily be visualized on a single page. Arena allows blocks to be grouped together to form sub-models, which is akin to the hierarchical structure Petri Nets allow. Again, the blueprint provided by the Petri Net provided a natural set of sub-models. The top-level view of the resulting Arena model is shown in Figure 3.20.

One "day" entity is created every 24 hours, and is then counted. As mentioned above, this count is assigned as an attribute to all the entities representing Petri Net tokens,

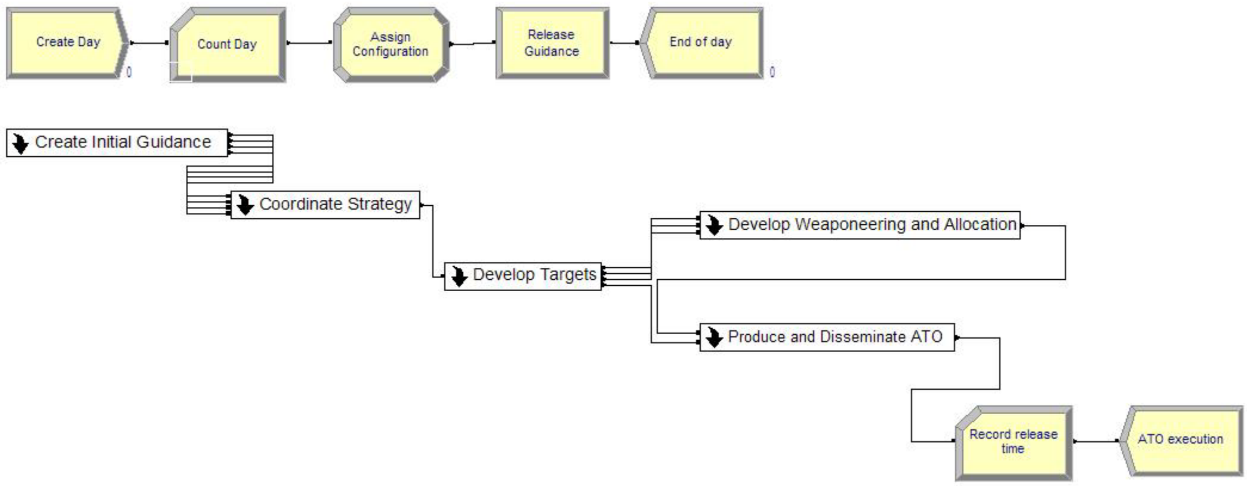


Figure 3.20 Top Level View of the Arena Implementation of the CPN

so that only those entities created on the same day can be batched together. The "Assign Configuration" block sets the parameters for the delay distributions, and assigns an overall configuration (i.e. all divisions collocated, AMD division reach back, etc.) to be modeled. This block is used in conjunction with the Arena Process Analyzer to rapidly run multiple configurations without having to manually change parameters between replications. The "Release Guidance" block sends a signal to the hold modules in the "Create Initial Guidance" sub-model. This releases the four tokens from their hold modules and begins the flow of information through the model. Figure 3.21 shows the Arena model built to accomplish this.

The remaining sub-models, which seek to mirror the functionality of the Petri Net, are considerably more complex. Each follows the same basic strategy in that entities representing all the tokens required by the sub-model are created and held at the start of model execution. This results in each sub-model containing a number of structures like those shown in Figure 3.22. In this example, which is a portion of the "Coordinate Strategy" sub-model, seven create blocks each create the tokens needed for the sub-

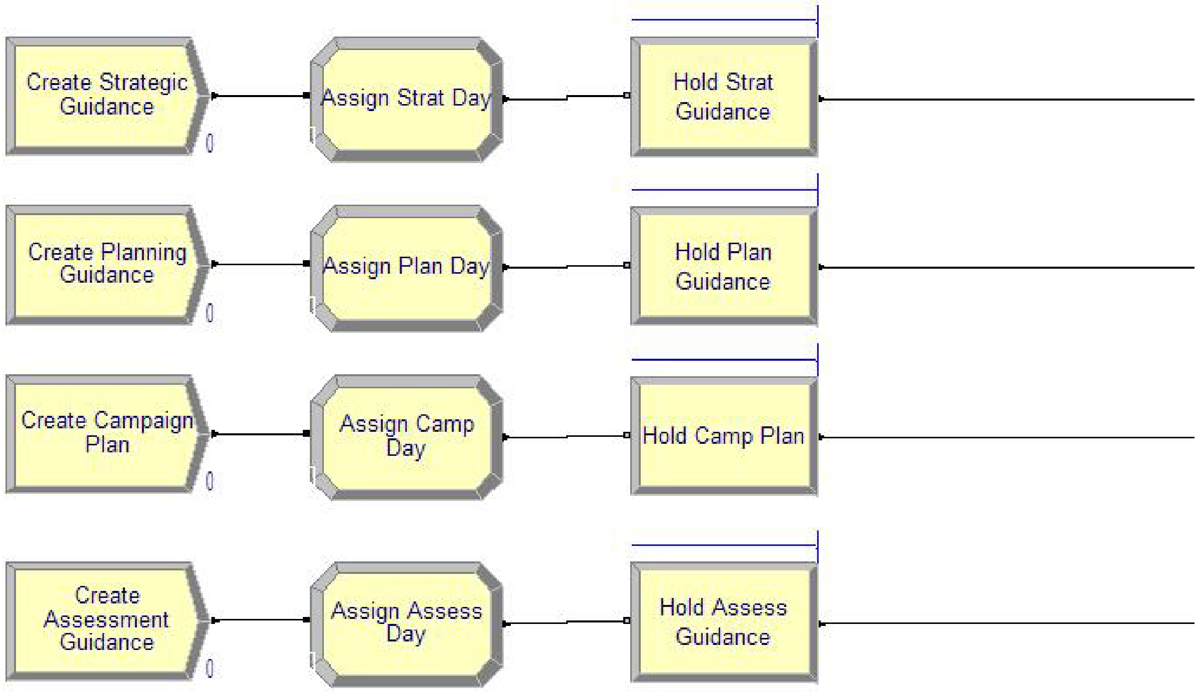


Figure 3.21 Creation of Arena Entities to Model CPN Tokens

model's execution. These entities are then available to be removed from their respective hold blocks at the appropriate time in the execution of the sub-model.

The entry of entities into a sub-model is analogous to tokens in the Petri Net arriving at the appropriate place to enable the firing of a transition. In the case of the "Coordinate Strategy" sub-model, this involves the four initial planning documents arriving. Figure 3.23, which represents the remainder of the "Coordinate Strategy" sub-model, shows this via the batch block labeled "Analysis Requirements". This batch block combines the four initial guidance documents into one entity which can then trigger the start of the "Analyze Mission" process. In our Petri Net once these tokens were in place, along with the tokens representing the necessary organizational resources, to allow the Analyze Mission transition to fire, the associated time delay would elapse and then the Mission

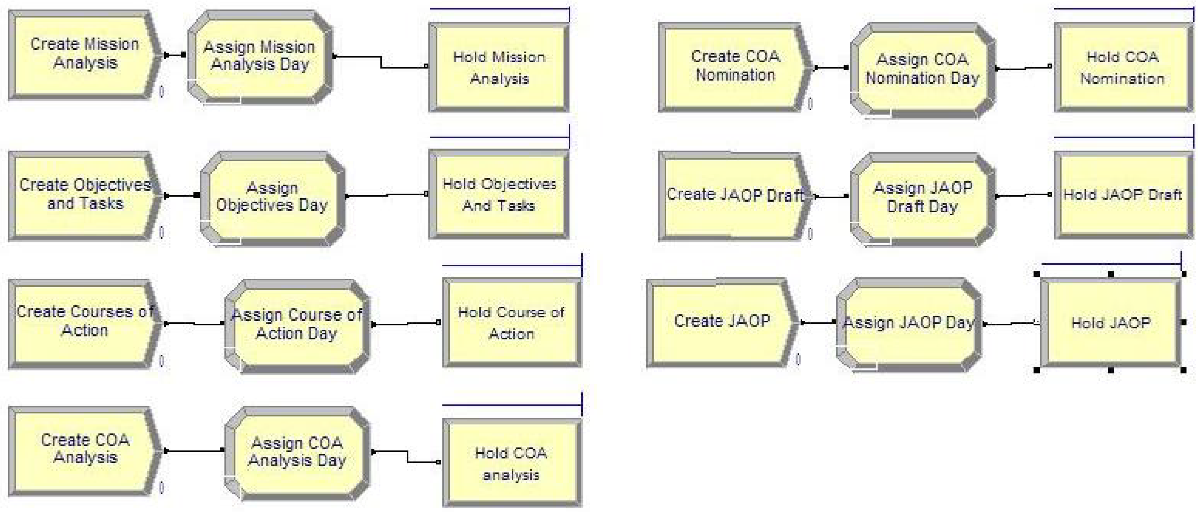


Figure 3.22 Token Creation for the Coordinate Strategy Process

Analysis token would be created. In our Arena implementation, the "Analyze Mission" process seizes the appropriate resources for a certain amount of time, then frees them and allows the batched entity representing the guidance documents to continue. This entity enters the "Remove Mission Analysis" block which removes an entity from the "Hold Mission Analysis" block. This is analogous to the Petri Net creating the Mission Analysis token. The Mission Analysis entity then proceeds to the "Determine Aerospace Objectives" process. The batched entity which represented the initial guidance documents is no longer required, and thus is separated back into its component entities and disposed of. This process repeats throughout the remaining processes within the sub-model. Finally, at the conclusion of the "Obtain Approval of Draft JAOP" process, an entity representing the approved JAOP is removed and passed out of the "Coordinate Strategy" sub-model and into the "Develop Targets" sub-model.

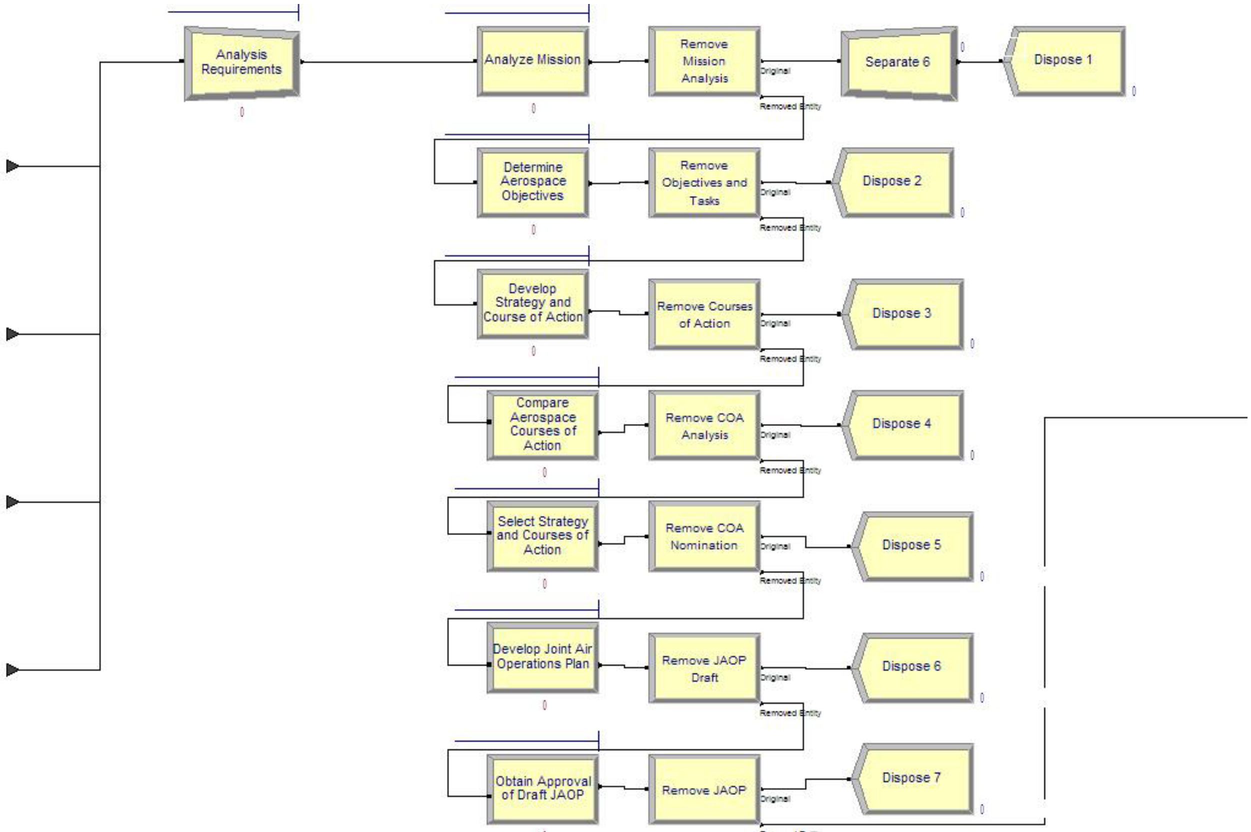


Figure 3.23 Arena Model of the Coordinate Strategy Process

The "Develop Targets" sub-model operates in much the same fashion as the "Coordinate Strategy" sub-model with one notable exception. Within the "Develop Targets" sub-model there are three instances in which the corresponding Petri Net places had multiple outgoing transitions. In order to model these in Arena, we used batched entities to represent the tokens and decide blocks to control their flow. A portion of the sub-model is displayed in Figure 3.24.

Here we once again see the structure whereby entities representing tokens are removed from hold modules, held for a certain delay, passed on to trigger the removal of the next entity, and then disposed of. What is new is the inclusion of separate and decide blocks to allow multiple exit paths from the processes. In order to achieve this, the requisite entities were created in pairs and batched prior to entering their hold modules. Upon their removal, they enter separate blocks, such as the one labeled "Split AOD

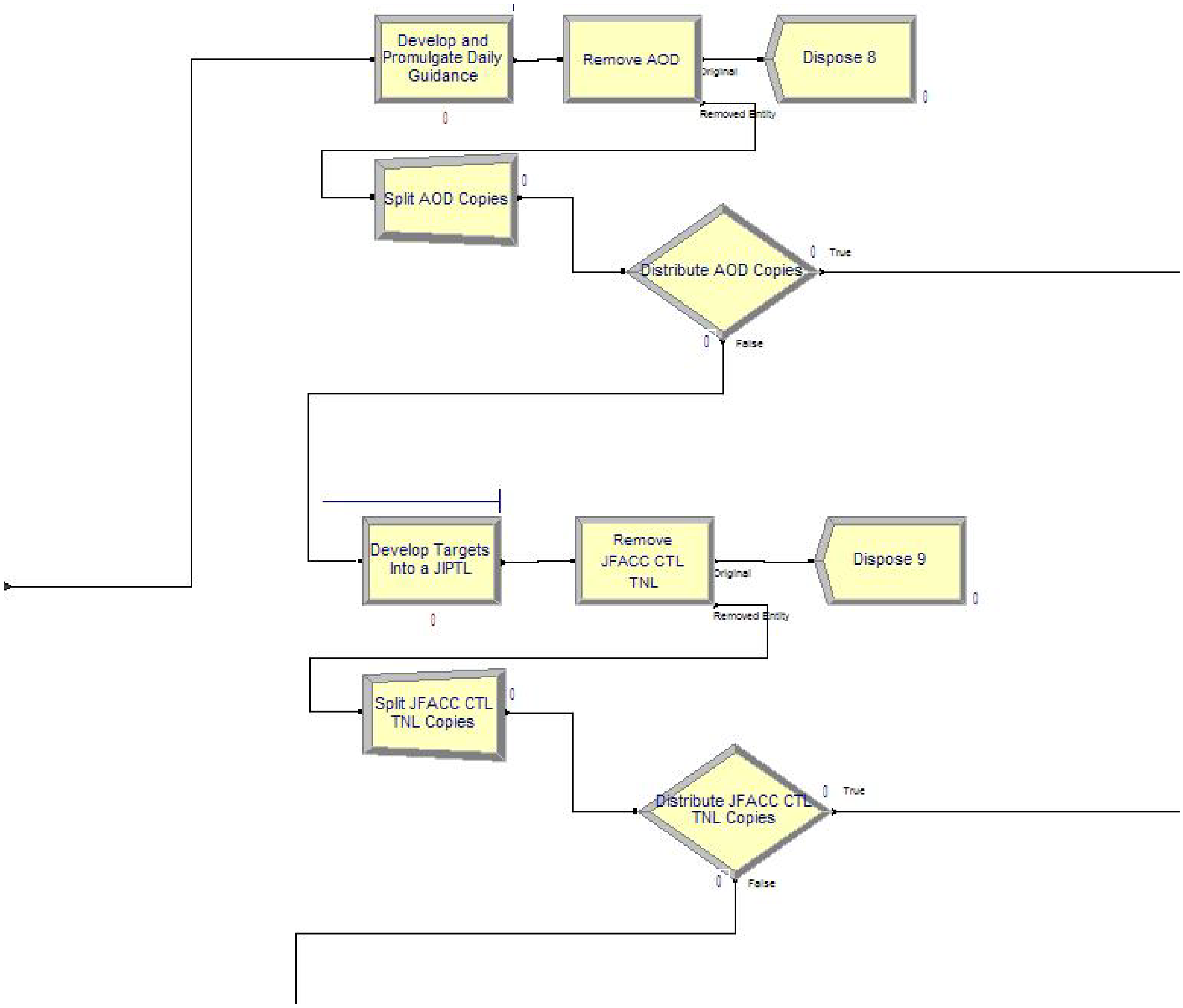


Figure 3.24 Arena Model of the Develop Targets Process

copies". They then enter a decide block which counts the number of entities that have gone down each path and makes sure that they are evenly distributed between the two. In the case of the AOD entity, one copy continues to the "Develop Targets into a JIPTL" block, while the other exits the sub-model and enters the first block of the "Develop Weaponeering and Allocation" sub-model. This captures the Petri Net construct where the place "AOD" has two outgoing transitions.

This methodology is pursued through the remaining sub-models. Finally, at the end of the "Produce and Disseminate ATO" sub-model, a decide block is encountered

with each path leading to an array of delay blocks. Up to this point, the behavior of our model has been completely deterministic. The amounts of time resources are seized and tokens are delayed within each of the process blocks have been fixed at the doctrinally determined nominal time. Lacking any real-world data on exact process performance, this standard was adopted in order to ensure some degree of validity. The drawback is that although the model has captured the flow of the ATO production thread, its behavior is completely deterministic up to this point. This is a divergence from the Petri Net model, which employed stochastic delays in the transitions. As will be described below, however, the overall functionality remains the same.

If one wanted to mimic the structure of the Petri Net in its treatment of delays, each of the connections between processes could be supplemented with a decide block with four exits, one for each of the configurations modeled. These exits would lead to a single delay block which held the entity for an amount of time drawn from the appropriate distribution. If the intent of the model were to simulate multiple, simultaneous ATO cycles this would be the best way to incorporate delays at the appropriate time in the model's execution.

In the case of our model, however, our intended use was much simpler: the modeling of a single ATO production cycle. At the level of abstraction we had chosen, we knew this to be a completely linear process with no competition for resources or other form of state-dependent delay. Thus, the point in the model execution at which a delay occurred had no bearing on the overall completion time of the ATO, only the value drawn for the delay mattered. This allowed us to simplify our model considerably by using a single decide block near the end of the model flow. Rather than having the various stochastic delays distributed throughout the model, they were grouped into a series of delays served by a single decide block. Our analysis of the Petri Net showed us exactly how many delays were associated with each of the four configurations, and so a long chain of delay blocks could be placed just prior to the disposal of the ATO entity and capture the same net effect as if they had been scattered along the transitions throughout the model.

Figure 3.25 shows the beginning of this delay chain. Following the decide block, which directs the ATO down one of four paths based on the value of the "configuration" global variable, an assign blocks generates 21 appropriate random draws and assigns them to attributes of the ATO entity. Each of the ensuing delay blocks then holds the entity for the amount of time specified in the appropriate attribute. The net effect is that the ATO's release time is 72 hours plus the sum of the randomly drawn delays. By having all configurations operate from the common 72 hour baseline, we see the net difference generated by their varying number of missed communication requirements. Of course, this behavior could easily be captured analytically without the use of a simulation model at all. By developing the basic process in an Arena model however, it is our intent to form a basis which can be built upon in later research aimed at capturing the concurrent nature of the true ATO process at a lower level of abstraction. This concept will be more fully explored in Chapter Five.

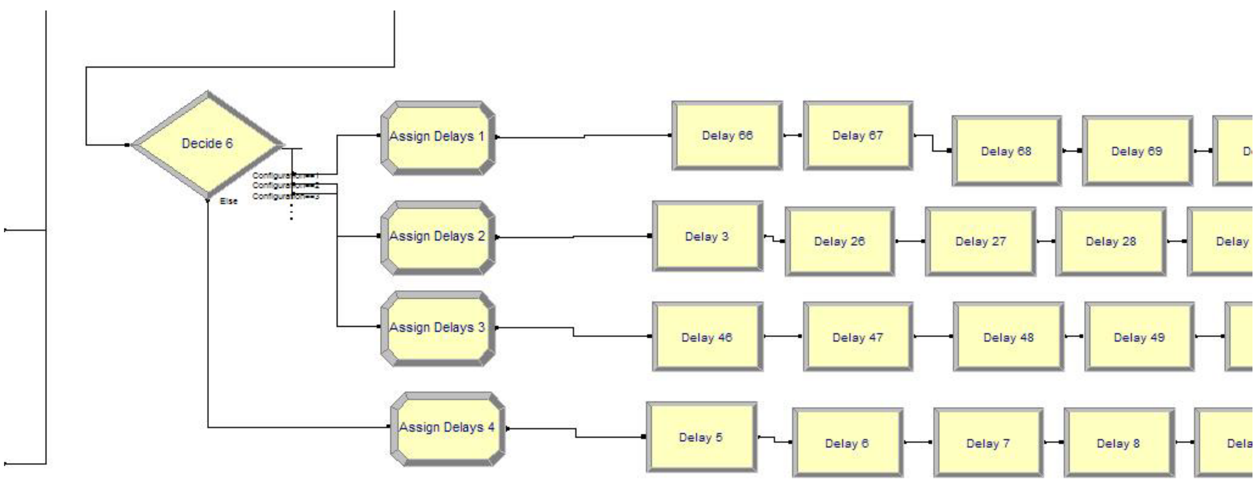


Figure 3.25 Process Delays in the Arena Model

3.5 *Executable Architecture Utilization*

The executable architectures were used to quantify the impact of the missed communication requirements. The CPN incorporated the rules that governed which distributions to use in each transitions. The same distributions were used in Arena. This captures the time penalty for missing at least one of the communication requirements pertaining to complexity, security, reliability, and speed. There are other aspects of the air tasking cycle that will be impacted by split operations and potentially cause further delays. The utilization of the executable models will be limited to addressing the impact of the communication requirements in Section 2.2. The process analyzer in Arena will be used to perform a sensitivity analysis of the distributions. This is important because the range or type of the distribution can not be verified without extensive research of AOC operations and the sources and durations of actual delays. In lieu of the inability to validate the distribution, the range of each distribution can be modified in order to find the critical ranges where the configurations are no longer statically different.

IV. Results

4.1 Communication Requirements Assessment

The earlier AFIT student group forerunning our research focused their research on the importance of grounding and effects of communication media. [16] Their research was concerned with information passing through four different domains: physical to information to cognitive, information to information, cognitive to information to information to cognitive, and cognitive to cognitive. [16, 35] Depending on the modality, different communication medium are acceptable to transmit the information. The communication mediums discussed were face-to-face, video teleconference, telephone, chat room, email, and bulletin boards.

There are 51 key processes that were also identified in the preceding research. These 51 processes were traced back to the OV-5 activities where the process is taking place. Since some of these processes are at higher levels of abstraction, some processes apply to multiple low level OV-5 activities. It is also true that more than process can apply to any given transition. This provided a mapping between the theoretical implications of communication theory discussed in Section 2.2 to the executable architecture of the air tasking cycle. This mapping enabled assessments to be made to determine the impact of communication media on split AOC operations.

From the four communication paths, characteristics of information were assessed. The characteristics selected for this study included complexity, security, speed, and reliability. These characteristics relate directly to the information that is passed throughout the air tasking cycle. The communication mediums previously mentioned have threshold levels for each characteristic, limiting the level of information that they are able to transmit. This provided a quantifiable way to assess the impact of communication requirements. When two divisions are not collocated, there is no possibility of face-to-face (F-T-F) communication. F-T-F communication contains the highest values for all characteristics in consideration as seen in Section 2.2. When F-T-F communication

is unavailable, the flow of information is dependent on other electronic communication methods. These other communication methods may not be sufficient to meet all of the requirements. The results have been consolidated in Table 4.1. This chart shows who is communicating and how many requirements where missed due to reliability or speed. Not all missed requirements apply to all configurations. The last four columns indicate what configurations did not meet the requirement for the transitions listed.

Seven of the 51 processes contained missed requirements. These seven processes impacted 21 of the 31 lowest level OV-5 activities for configurations 2 and 4 while impacting 20 activities for the remaining two configurations. This demonstrates how requirements that are at a higher level of abstraction can impact multiple low level activities. Only two characteristics were not met: reliability and speed. Reliability was by far the dominant failure mode with 32 missed requirements. Speed only had 17 missed requirements. The power of this analysis is that it reveals not only failure modes, but exposes the number of missed requirements at the lowest level of OV-5 decomposition. These number of missed requirements are shown in Table 4.2.

Logically, the baseline has the least amount of missed requirements with a total of 41. This discloses that there are some communication paths used that are not able to utilize face-to-face communication. These areas provide insight to areas that can be further examined to improve the current process from the communication modality perspective. Also it is reasonable that configuration four has the most missed requirements with a total of 49.

Originally, this analysis was intended to be done using CPN Tools, but as discussed in Section 3.2 software limitations made this infeasible. With a different CPN software package which has more flexibility in the code segment, this type of analysis could easily be implemented. In lieu of this hurdle, the requirement assessment was used as an input to the executable architectures in order to quantify their temporal impact.

Table 4.1 Information Exchange Results

Tranisition	Source	Destination	Reliability Difference b/n Req't and Available	Speed Difference b/n Req't and Available	Configuration 1	Configuration 2	Configuration 3	Configuration 4
A11	AMD (AMCT)	JFACC, AOC director	1	0		X		X
A21	ISR (PED)	Combat Plans	1	0	X		X	X
A21	AMD (AMCT)	JFACC, AOC director	1	0		X		X
A311	ISR (PED)	Combat Plans	1	0	X	X	X	X
A311	ISR (TGT/CA)	Combat Plans	0	1	X	X	X	X
A312	ISR (PED)	Combat Plans	1	0	X	X	X	X
A312	ISR (ACFT)	ISR Chief	0	1	X	X	X	X
A312	ISR (TGT/CA)	Combat Plans	0	1	X	X	X	X
A313	ISR (PED)	Combat Plans	1	0	X	X	X	X
A313	ISR (ACFT)	ISR Chief	0	1	X	X	X	X
A313	ISR (TGT/CA)	Combat Plans	0	1	X	X	X	X
A314	ISR (PED)	Combat Plans	1	0	X	X	X	X
A314	ISR (ACFT)	ISR Chief	0	1	X	X	X	X
A314	ISR (TGT/CA)	Combat Plans	0	1	X	X	X	X
A315	ISR (PED)	Combat Plans	1	0	X	X	X	X
A315	ISR (ACFT)	ISR Chief	0	1	X	X	X	X
A315	ISR (TGT/CA)	Combat Plans	0	1	X	X	X	X
A316	ISR (PED)	Combat Plans	1	0	X	X	X	X
A316	ISR (ACFT)	ISR Chief	0	1	X	X	X	X
A316	ISR (TGT/CA)	Combat Plans	0	1	X	X	X	X
A317	ISR (PED)	Combat Plans	1	0	X	X	X	X
A317	ISR (ACFT)	ISR Chief	0	1	X	X	X	X
A317	ISR (TGT/CA)	Combat Plans	0	1	X	X	X	X
A321	ISR (TGT/CA)	Combat Plans	0	1	X	X	X	X
A322	ISR (TGT/CA)	Combat Plans	0	1	X	X	X	X
A323	ISR (TGT/CA)	Combat Plans	0	1	X	X	X	X
A324	ISR (TGT/CA)	Combat Plans	0	1	X	X	X	X
A41	ISR Platforms	ISR (ACFT)	2	0	X	X	X	X
A41	JFACC Intent	Plans Div	3	0			X	X
A42	ISR Platforms	ISR (ACFT)	2	0	X	X	X	X
A43	ISR Platforms	ISR (ACFT)	2	0	X	X	X	X
A44	ISR Platforms	ISR (ACFT)	2	0	X	X	X	X
A45	ISR Platforms	ISR (ACFT)	2	0	X	X	X	X
A46	ISR Platforms	ISR (ACFT)	2	0	X	X	X	X
A46	JFACC Intent	Plans Div	3	0			X	X
A47	ISR Platforms	ISR (ACFT)	2	0	X	X	X	X
A48	ISR Platforms	ISR (ACFT)	2	0	X	X	X	X

4.2 CPN Analysis

Thirty runs were made for each configuration and the time means and 95% confidence intervals were calculated. These results are shown in Table 4.3. Real numbers

Table 4.2 Configuration Results

Transition	Configuration 1	Configuration 2	Configuration 3	Configuration 4
A11	0	1	0	1
A21	1	2	1	2
A311	2	2	2	2
A312	3	3	3	3
A313	3	3	3	3
A314	3	3	3	3
A315	3	3	3	3
A316	3	3	3	3
A317	3	3	3	3
A321	1	1	1	1
A322	1	1	1	1
A323	1	1	1	1
A324	1	1	1	1
A41	2	2	5	5
A42	2	2	2	2
A43	2	2	2	2
A44	2	2	2	2
A45	2	2	2	2
A46	2	2	5	5
A47	2	2	2	2
A48	2	2	2	2
Total	**41**	**43**	**47**	**49**

are not allowed in CPN Tools, so integer values for minutes had to be used. Minutes were chosen as the time units since it would not be logical to have delays in the magnitude of hours at our level of decomposition. For example, if hours were chosen as the primary time unit each lowest level activity would be delayed for at least an hour depending on the distribution range. There are a number of activities that are completed within an hour and it would not make sense to double or triple the activity time based on evaluating communi-

cation media. The results will be converted to hours after the model has been verified to be functioning properly.

Table 4.3 CPN Results (Minutes)

	Mean	Variance	CI	Low Bound	High Bound
Configuration 1	4442.83	108.97	3.74	4439.10	4446.57
Configuration 2	4656.37	1067.14	11.69	4644.68	4668.06
Configuration 3	4695.73	1570.69	14.18	4681.55	4709.92
Configuration 4	5090.00	4915.66	25.09	5064.91	5115.09

In order to verify the CPN is functioning properly the average ending time and variance must be checked. The mean time for each configuration for the 30 runs has been shown in the first column in Table 4.3. These can be verified by creating a 95% confidence interval about the mean and checking to see if the confidence interval contains the true mean. In most systems the theoretical mean is not known, but in our system it can be calculated since the distribution is known that created the random time delays. This process will be presented for the baseline configuration.

One of the assumptions in Section 3.1 is that the minimum time to produce an ATO is 72 hours. This is equivalent to 4,320 minutes. The average delay time for a discrete distribution with a range of one to five minutes is simply three minutes. Using the configuration results in Table 3.2 as a reference, there are five transitions that missed one requirement, nine transitions that missed two requirements, and six transitions that missed three requirements. The following equation is used to determine the average delay:

$$E[c * g(Y)] = cE[g(Y)]$$

This equation means that if the average delay is three minutes and is multiplied by two, in the situation where two requirements where missed, the average delay would be six minutes. This allowed the theoretical average delay time to be calculated as follows:

$$5(transitions\ with\ one\ missed\ req't) * 3(average\ delay) + 9*6+6*9 = 123\ minutes$$

Notations have been added around the first time to explain what the terms represent. The delay of 123 minutes is added to the static 4320 (72 hour) processing time for a theoretical value of 4443 minutes. As can be seen in Table 3.3, this is very close to the sampled mean time of 4442.83 minutes. The theoretical mean for configurations two through four are 4,664 minutes, 4,696 minutes, and 5,079.5 minutes respectively. All of these values fall within the 95% confidence intervals that have been calculated in Table 4.3.

In calculating confidence intervals, an assumption must be made that the data is normal. Even though discrete distributions are being used over a relatively small interval, the sample space does become normal by adding multiple discrete random variables. This can be visualized by Figure 4.1. Five minute bins have been constructed for the sampled data for the baseline configuration. This is the configuration with the least amount of variability so by illustrating that this data is mound shaped, the other configurations will also be validated. The distribution of the sample points appears to follow the bell-shaped pattern of a normal distribution. Also, with 30 sample points, the central limit theorem can be invoked. This allows the assumption of normality to be made regardless of the distribution of the population from which the sample is taken.

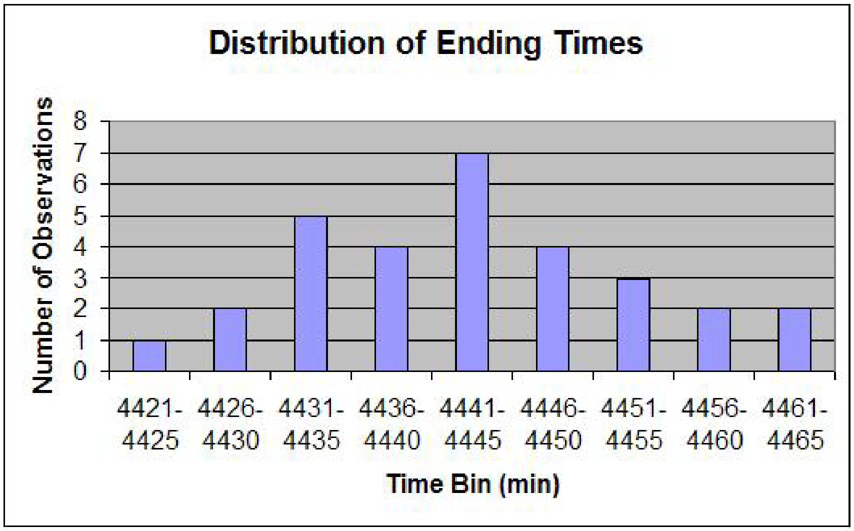


Figure 4.1 Configuration One Distribution

The last thing that needs to be checked to verify that the results of CPN Tools are within expectations is the variance of the sampled data. Once again, since the originating distributions are known, the theoretical variance of the system can be found. The formula for variance is shown below:

$$Variance = \sigma^2 = E(Y^2) - \mu^2$$

$$where\ \mu\ is\ the\ mean$$

$$E(Y^2) = \sum_y y^2 p(y)$$

$$where\ y\ is\ the\ sampled\ value\ and\ p(y)\ is\ the\ probability$$

The mean of a discrete distribution with a range from one to five is already known to be three. The expected value of y^2 is simply: $\sum_{y=1}^{5} y^2 p(y)$. In our case, this yields an expected value for y^2 of 11. The variance of a discrete distribution with a range from one to five is two. One more equation is needed in order to find the variance of the situations where a constant is multiplied by a random probability function.

$$Var(cX) = E(cY - E(cY))^2 = c^2E(Y - E(Y))^2 = c^2Var(X)[14, 14]$$

This equation proves that the variance is changed by the square of the constant. Using this equation, in the case where there are two missed requirements, the variance would be multiplied by two squared, or four. This allows the variance to be calculated for each transition. Since the variance is an additive property, the sum of the variances would be the variance of the system. The variance of the baseline has been found to be 190. The variance of the remaining three configurations are 1848, 2557, and 10563, respectively.

Now that the theoretical variances are known as well as the sampled variances, 95% confidence intervals can be constructed around the sampled variances to determine if they contain the theoretical variances. The formula used to calculate the confidence intervals for the sampled variances is shown below. [28, 408]

$$\left(\frac{(n-1)S^2}{X^2_{\alpha/2}}, \frac{(n-1)S^2}{X^2_{1-\alpha/2}} \right)$$

where: n is the number of sample points,

S is the sample variance,

α is the selected confidence level,

and X^2 is the appropriate chi-square statistic

The results of the confidence intervals are shown in Table 4.4. All of the confidence intervals hook the theoretical variance, except for configuration four. The fourth configuration has the largest theoretical variance of all of the configurations. This also would imply that the population (ie, possible outcomes of time delays) of the fourth configuration is significantly larger than the other three configurations. With only thirty sampled points it is very likely to not select extreme data points. The theoretical variance is calculated based upon the complete population, including the unlikely extreme data points. This

explains the trend for the theoretical variance being higher than the observed variance. Since the same model was used for all four configurations and only configuration four did not capture the theoretical mean, the assumption that the model is behaving properly will be made.

Table 4.4 95% Variance Confidence Intervals

	Variance	Low Bound	High Bound	Theoretical Value
Configuration 1	108.97	69.12	196.93	190.0
Configuration 2	1067.14	676.85	1928.51	1848.0
Configuration 3	1570.69	996.23	2838.52	2557.3
Configuration 4	4915.66	3117.82	8883.49	10563.3

Now that the CPN has been demonstrated to be performing as expected the results will be further analyzed. Table 4.5 converts the results to hours. The confidence interval is still 95%. It is important to verify the model before converting to hours because constants have a nonlinear impact on variance. By converting to hours by dividing by a constant would make checking the theoretical variance more complicated.

Table 4.5 CPN Results (Hours)

	Mean	Variance	CI	Low Bound	High Bound
Configuration 1	74.05	0.03	0.06	73.99	74.11
Configuration 2	77.61	0.30	0.20	77.41	77.81
Configuration 3	78.26	0.44	0.24	78.02	78.50
Configuration 4	84.83	1.37	0.42	84.41	85.25

The same confidence intervals that were calculated around the sample means can be used to determine if the configurations are significantly different. If the confidence intervals overlap for any two configurations, those configurations can be viewed as the same from a statistical perspective. As can be seen from Table 4.5, all the configurations are significantly different, although configurations two and three are very close to overlapping. They are close because they use the same distribution (discrete from 1 to 15) and are different because configuration three has four more missed requirements. Arena was used to find at what points configurations that do not use the same distribution become statistically the same. These are the critical points of the distribution.

4.3 Arena Analysis

4.3.1 Ensuring Model Equivalence. We began by performing a similar analysis to that described above to first verify that our Arena model was indeed mirroring the performance of the Petri Net. Since the models use different underlying random number generators, results of identical configurations could be expected to differ. The overall behavior of the models could be expected to agree however, particularly regarding their proximity to the theoretical mean and variance. Thirty replications were used in the Arena model as well to ensure sample size was not a cause for discrepancies. As shown in Table 4.6 below, the 95% confidence interval which Arena constructs automatically captured the theoretical mean for all four configurations. Applying the same method as above to construct confidence intervals around the variance calculations, we found that again the 95% confidence interval captured the theoretical value in all but the fourth configuration. The results are summarized in Table XX below.

To further compare the performance of the two models and ensure their performance was not statistically different, we formed a 95% confidence interval around the difference in their means. To ensure our accuracy we employed both a traditional Paired-*t* test and a Modified Two-Sample-*t* test which Law and Kelton attribute to Welch. [11, 557-559]

4.3.1.1 Traditional Paired-t Test. Configuration One's results were selected for both the CPN and Arena models. This configuration was used in a traditional paired-t test as well as a modified two sample-t test. For the traditional paired-t test, we begin by defining $\zeta = \mu_1 - \mu_2$ where μ_1 is the expected ATO cycle produced by the Arena model and μ_2 is the corresponding response from the Petri Net in the baseline configuration. We then pair the observations for ATO cycle time, with X_{1j} representing the j (in this case 30) values realized by the Arena model and X_{2j} representing the values realized by the Petri Net. This allows us to form 30 values for a new variable $Z_j = X_{1j} - X_{2j}$ for $j = 1, 2, \ldots, n$. This new variable is now Independent and Identically Distributed and $E[Z] = \zeta$. A confidence interval for ζ can then be formed by letting

$$\overline{Z}(n) = \frac{\sum_{j=1}^{n} Z_j}{n}$$

and

$$\widehat{Var}[\overline{Z}(n)] = \frac{\sum_{j=1}^{n} [Z_j - \overline{Z}(n)]^2}{n(n-1)}$$

and form the confidence interval

$$\overline{Z}(n) \pm t_{n-1,1-\frac{\alpha}{2}} \sqrt{\widehat{Var}[\overline{Z}(n)]}$$

Applying this process to the data generated by the two models results in a 95% confidence interval, in minutes, of

$$-4.62 \pm 36.596 = [-41.2, 31.97]$$

Since this interval covers 0 we can conclude at the $\alpha = 0.05$ level that there is insufficient evidence to say that these two models are significantly different.

The paired-t test is best employed in situations where the simulation experiment has been designed such that there is some degree of correlation between the output of the two models. The use of common random numbers (CRN), in which the same random number streams are used in both models, is a common method of achieving this. This results in a reduction in the size of $Var(Z_j)$ and thus in the width of the confidence interval.

4.3.1.2 Modified Two-Sample-t Test. Since our two configurations were being modeled on two different simulation platforms, it was not possible to construct a CRN experiment. Thus, as a final check of the equivalence between the two models, we performed a modified two-sample-t test. The traditional two-sample-t test requires that the variance of the two samples be the same, or the validity of the calculated confidence

interval suffers. Since we were using two different simulation platforms with two different random number generators, a far safer assumption is that the population's variances are not equal. The modified test does require independence between the data sets, but the use of two simulation platforms makes this a safe assumption.

The modified test begins in the usual manner by defining

$$\overline{X}_i(n_i) = \frac{\sum_{j=1}^{n_i} X_{ij}}{n_i}$$

and

$$S_i^2(n_i) = \frac{\sum_{j=1}^{n_i}[X_{ij} - \overline{X_i}(n_i)]^2}{n_i - 1}$$

for $i = 1,2$. We then compute the *estimated* degrees of freedom

$$\hat{f} = \frac{[\frac{S_1^2(n_1)}{n_1} + \frac{S_2^2(n_2)}{n_2}]^2}{\frac{\frac{[S_1^2(n_1)]^2}{n_1}}{(n_1-1)} + \frac{\frac{[S_2^2(n_2)]^2}{n_2}}{(n_2-1)}}$$

and form a confidence interval

$$\overline{X}_1(n_1) - \overline{X}_2(n_2) \pm t_{\hat{f},1-\frac{\alpha}{2}} \sqrt{\frac{S_1^2(n_1)}{n_1} + \frac{S_2^2(n_2)}{n_2}}$$

as an approximate $100(1-\alpha)$ percent confidence interval for ζ. As expected, this yielded a non-integer value for $\hat{f}$ so it was necessary to use MatLab in order to calculate the appropriate t value. This ultimately led to a 95% confidence interval on the difference in the means of

$$-4.62 \pm 34.067 = [-38.69, 29.44]$$

This so-called *Welch confidence interval* [11, 559] again covers 0 so we again conclude that at the $\alpha = 0.05$ level there is insufficient evidence to say that these two models are significantly different.

4.3.2 Arena Results. At this point we were very confident that our Arena model was accurately mirroring the performance of our Petri Net for all configurations. This allowed us to utilize the Arena Process Analyzer (PAN) to rapidly test multiple parameter settings across the configurations. One of the weakest points in our model of the ATO development thread, we felt, was the discrete distributions used to determine the delays associated with missed communication requirements. The ranges were chosen because they seemed reasonable, making them one of the only elements of the model not directly traceable to an element of the validated architecture. While we knew that we could not validate these parameter settings, we felt it might provide some insight to investigate the impacts of varying their values.

Figure 4.2 below shows the nine scenarios PAN was configured to execute. The controls to be modified were the maximum values of the delay distributions. Values ranging from the baseline of 15 to a maximum of 30 were defined in increments of five for both configuration two and three. Configuration four was run in its baseline configuration with a maximum delay of 30. Of interest was whether any of these configurations would result in overlapping confidence intervals for the predicted mean ATO cycle time.

Table 4.6 displays the numerical results of the PAN runs, while Figure 4.3 presents the data graphically. As Figure 4.3 makes clear, there was only one overlap in the 95% confidence intervals on the predicted mean ATO cycle time. Configuration three, in which the Combat Plans and Strategy Divisions are at reachback locations overlaps with configuration four, which further distributes the Air Mobility Division. This overlap occurs only when configuration three's maximum delay is increased to 30 minutes, which was the highest value tested.

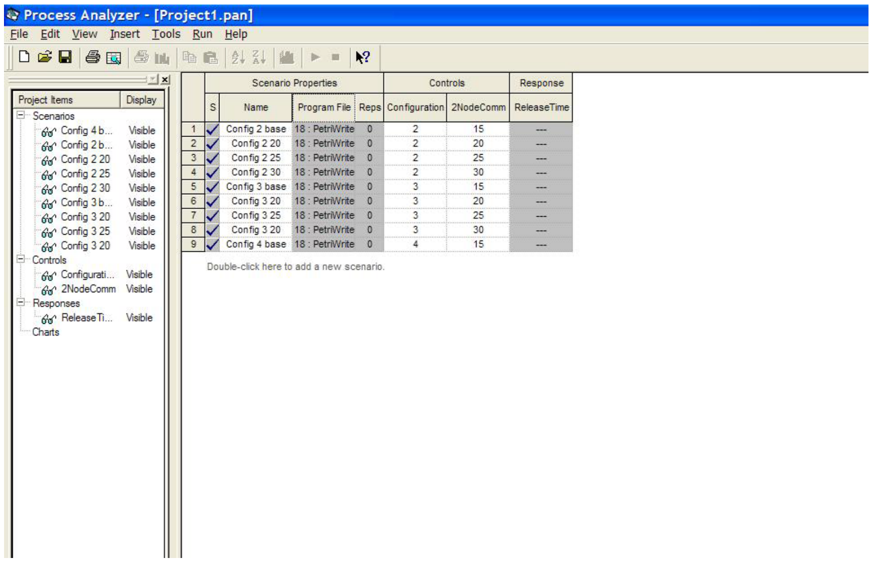


Figure 4.2 PAN Scenario Definition

Table 4.6 95% Confidence Intervals on Mean ATO Cycle Time

	Mean	Variance	Low Bound	High Bound	Theoretical Mean
Baseline	4438.206	179.93	4433.406	4443.006	4443
AMD at TACC	4654.188	1124.53	4642.188	4666.188	4664
CPD & SD Reachback	4687.746	1487.19	4673.946	4701.546	4696
AMD at TACC, and CPD & SD Reachback	5090.088	5692.94	5063.088	5117.088	5079.5

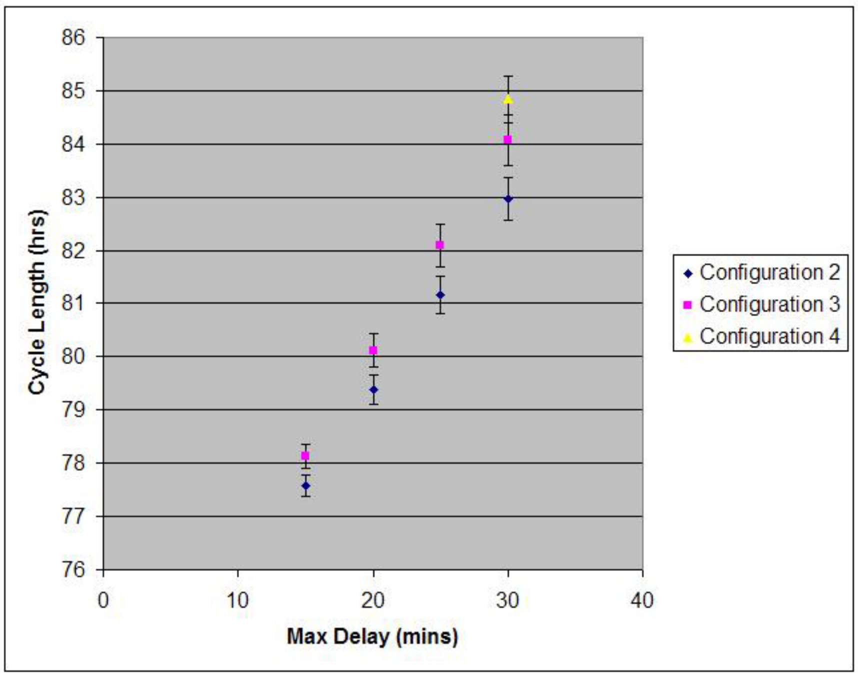


Figure 4.3 PAN Confidence Intervals on Predicted Mean ATO Cycle Time

V. Conclusions and Recommendations

5.1 Conclusions

5.1.1 Significance of Executable Architectures. The process of utilizing DoD Architecture Framework (DoDAF) views as the foundation to build an executable architecture provided many insights to the analysis of distributed air operations. A common executable architecture was simulated in CPN Tools and Arena. While each of these tools provided distinctly different capabilities and limitations, the foundation was the structure produced in the static DoDAF design.

5.1.1.1 CPN Implementation. The primary strength of CPNs is that it provides a distinct and undeniable ability to trace each component of the Colored Petri Net back to a DoDAF product. This traceability is ideal for validating and creating a pedigree for an executable model. The relationship between the executable architecture and originating DoDAF products is so strong that a properly constructed CPN could be used to create a complete OV-5, OV-6a, and all of the entities of the OV-7. This linkage enables anyone with a working knowledge of DoDAF to easily create and read CPNs.

CPN Tools was found to be extremely limited in the ability to do analysis. This is due to software limitations within CPN Tools. The methodology of employing CPNs is substantially more powerful than the implementation within CPN Tools. Other tools have been researched and also come with their own unique limitations.

There were three obvious tool limitations within CPN Tools. These were the ability to generate batch runs, overly constricted code segments on transitions, and the prohibition of real numbers. The ability to create batch files with varied inputs would allow higher methods of analysis such as a design of experiments or sensitivity analysis to be used. This would be a great compliment to the current features of being able to generate a simulation and state space report. The limitation that was encountered with the code segment was the limit of a single “IF-THEN-ELSE” construct. “IF” statements can be nested, but this

requires that all possibilities be enumerated. For example, if it is desired to check to see if three tokens have the same location, seven nested if statements would be required. This could be simplified by allowing more than one if statement within a code segment. The biggest tool limitation is the prohibiting of real numbers. This was a feature that was lost when converting from the older version Design/CPN to the latest version of CPN Tools. This prohibits any random distributions to be used that are not discrete as well as severely impacting the ability to use mathematical equations to modify and assess MOEs.

5.1.1.2 Arena Implementation. As mentioned above, CPN Tools presents some limitations in its ability to run a simulation experiment involving multiple replications and the resulting statistics. CPN Tools predecessor, Design/CPN, possessed a more robust simulation engine which allowed multiple replications to run in a batch mode and automatically gathered multiple statistics across the replications. Unfortunately, Design/CPN is no longer supported and does not allow hierarchical nets. These were two of the primary considerations behind our decision to use CPN Tools.

In order to supplement CPN Tools simulation capabilities, a parallel model was developed in Arena. As described in Section 3.4.2 this was a large, and somewhat cumbersome model. Arena's intended purpose as a modeling tool for business or manufacturing processes does not immediately lend itself to the logical constructs of a Petri Net. This resulted in the large number of unique entities that were developed to mirror the Petri Net tokens.

An Arena model developed from scratch to model the ATO production thread would likely have been very different. It likely would not have displayed the traceability back to DoDAF products that the Petri Net displayed. The process of mirroring the Petri Net once it was developed however was relatively straightforward. The Petri Net provided the blueprint as detailed in Section 3.4.2. Although the resulting net lacked the "elegance" of the Petri Net, it was developed rapidly and easily captured the same performance.

Once developed, multiple replications of a variety of configurations could be run in a batch mode with the Arena Process Analyzer. Relevant statistics were gathered and 95% confidence intervals were generated automatically. Equivalent processing using CPN Tools would have been a very lengthy process involving manual intervention between the runs of differing configurations. This was the real strength of the Arena model and the primary reason we chose to construct it.

5.1.2 Interpretation of Numerical Results. As mentioned in Chapter 4 we began by comparing each model individually to the theoretical values we should expect to see. Once this was done, we then compared the two models to one another to ensure that their performance was equivalent. We observed behavior from both models consistent with the theoretical values and found that a 95% confidence interval on the difference between their means covered zero. This led us to conclude that the models were both performing properly on their own and in comparison to one another.

We then turned to comparing the original four configurations. The most obvious observation was that the 95% confidence intervals for the predicted ATO cycle time did not overlap with any of our four configurations. This was to be expected, though we did think that configurations two and three were similar enough that an overlap might have been possible, since they both drew from a discrete distribution with the same bounds.

After running PAN to compare a variety of configurations we found that the only overlap in confidence intervals occurred between configurations three and four when configuration three's maximum bound was increased to 30 minutes. From this we concluded that the various configurations are not overly sensitive to small changes in the bounds of the discrete distributions. The fact that configuration three's maximum bound had to be doubled before an overlap was observed supported this conclusion. This condition also represented a violation of assumption four from Section 3.1.2, namely that the number of communicating nodes will impact the range of the delay distribution. In this case the overlap did not occur until configuration three's maximum was increased to

30 which is the same value as configuration four's maximum. Since configuration four represents communication between one more node than configuration three, this is not consistent with our assumptions.

The behavior of the 95% confidence intervals as the distribution bounds were increased also supported the use of our simulations as a comparitive tool. Due to the fact that we were unable to validate the bounds of the distributions used to determine delay values, neither the CPN or Arena model may be used to predict absolute performance of a configuration. However, since their behavior remains statistically different within the bounds of our assumptions, their relative performance as measured by either tool is a useful comparative value.

The results of our analysis produced a rank ordered list of the configurations. This list is shown in Table 5.1. The list is ordered from the best, or lowest mean time to produce the ATO, to the worst. Coincidently, the order that the configurations were numbered is the same as their ranking. The CPN mean times have also been shown to provide a quantifiable measure of the differences. The driving factor of our model was the distribution range. This can be deducted by the overlap in the confidence intervals that was discovered between configurations three and four. Even though configuration four had two additional missed requirements than configuration three, they were found to be statistically the same when they used the same distribution range of one to thirty minutes.

Table 5.1 Ranked Configurations

Rank	**Configuration**	**CPN Mean Time (Hrs)**
1	One	74.05
2	Two	77.61
3	Three	78.26
4	Four	84.83

5.1.3 Lessons Learned. Throughout the course of this project, many things were learned pertaining to developing and assessing executable architectures. Early in the process it is essential to have the problem well defined and the measures of effectiveness

selected. Executable architectures can be utilized to analyze many different attributes. Different attributes will affect how the rules are developed and implemented. Significant time will be lost by not having a well defined problem, direction, and goal, as well as a firm method of measuring progress toward that goal.

An important aspect of the problem definition and MOE selection process is ensuring a methodology is in place for capturing the relevant information. Throughout this project we grappled with issues such as how to define the quality of an ATO or quantify the impact of a change in communication media on the quality of the information passed. Though these are central concepts to evaluating the effectiveness of any AOC configuration, they are difficult to quantify much less predict without executing the processes via an AOC exercise or some other physical implementation of the architecture. A simulation is only as good as the data used in its construction, and where that data is un- or ill-defined the best simulation is little more than an educated guess.

With that said, a simulation can still provide insight into the relevant processes. By identifying bottlenecks or uncovering other emergent behavior, a model can reveal aspects of the system that were not apparent in a static analysis. So long as these behaviors are not strictly a result of inaccurate parameter settings, they provide hitherto unknown information about the behavior of the system.

It is equally important to learn as much as possible about a software tool before using it for model development. As previously mentioned, this project used CPN Tools to develop the executable architecture in a colored petri net. While time was spent to assess various tools at the onset of the project it was infeasible with the time and manpower constraints to foresee all potential shortcomings of CPN Tools. In hindsight, Design/CPN might have provided greater functionality, but had a much greater acquisition time since it required mailing request forms to Denmark and waiting for the software to be mailed back and then approved for use through AFIT/SC. While it is speculated that Design/CPN would provide greater functionality, it should be noted that this is based upon reading

about the program and not on experience. Design/CPN would undoubtedly come with its own unexpected limitations.

5.2 Recommendations

The research of this project provides a great springboard for further endeavors in creating and assessing executable architectures. Most recommendations are specific to expanding upon the research presented within this thesis and other recommendations are in general areas that have not been fully explored.

5.2.1 Project Specific Recommendations. There are a number of assumptions that were made in order to scope this problem to a size that could be addressed with the time and manpower available. One of these assumptions was the focus of developing the ATO, which is a subset of the whole air tasking cycle. Completing this air tasking cycle would provide a more detailed analysis. The two remaining steps that would need to be decomposed to complete this cycle are execution and assessment. By incorporating these two remaining steps new MOEs could be identified that pertain to the execution and assessment of actual MOEs. These MOEs could impact the development of the next ATO since the process is a sequential cycle.

Another assumption was to model only one ATO, when in reality there are four ATOs that are being developed simultaneously. This concurrent development process should be modeled and examined. This would cause the state space to be "nonlinear" since it would not be specified in which order the four ATOs would fire. More importantly it would allow bottlenecks to be identified if random delays are still incorporated. If four ATOs are modeled and are simulated for multiple cycles, there may be specific steps of the air tasking cycle that are sources for bottlenecks. This would allow the most constraining phase of the air tasking cycle to be identified and potentially rearchitected in order to maximize throughput of ATOs.

The organizational resources for this model were implemented at the division level. This high level of abstraction limits split operation flexibility. There may be other configurations that are more suitable for the split operations concept and should be explored. Modeling resources at the team level with four configurations could also show where there are resource constraints due to a team being involved with multiple phases of ATO development. This is especially true for the ISR division. Modeling organizational resources at the division level also constrained the potential to run simultaneous ATO cycles. Since the divisions were not decomposed into teams that could be allocated across different processes at the same time, no parallel processing could take place.

Time was utilized as the MOE to assess communication modality of various AOC configurations. Pursuing quality as an MOE would provide further insight on how split operations effect the air tasking cycle. Quality would help assess the value of the information that passes throughout the air tasking cycle. Since the information is passed between divisions and throughout the whole air tasking process it would seem logical that the quality is not an independent attribute. This would mean that a low quality value of one document would impact any related documents that were subsequently developed. This residual effect of assessing quality throughout the ATO is intriguing.

5.2.2 General Recommendations. The tight interdependence between a CPN and DoDAF products has been displayed in this project as well as many other publications. With this tight dependency, an automated method could potentially be developed that describes how CPNs should be used to modify an architecture. There may be more than one way that a CPN might provide feedback into the static architecture. For example, valid process flow can be determined by simulating a CPN. The sequential nature of the activity model could be verified through this simulation. A less obvious feedback could be through assessing MOEs. If a MOE is carefully selected where it will either pass or fail at the lowest level activities of the OV-5, it could be used to highlight areas of the architecture that might need to be reevaluated. This area of utilizing a CPN to provide feedback to the static architecture is not well defined and should be pursued.

The coherency between static architecture products and a CPN is ideal for specifying an automated process for developing an executable architecture. The process of taking an OV-5, OV-6a, OV-6b, and OV-7 to create an executable architecture has been well documented. It would be logical to create a tool to automate this process. This was examined briefly, but had to be abandoned due to time constraints. One interesting approach was the potential to partially automate the development of CPN models from a static architecture OV-5. CPN Tools utilizes the extensible markup language (XML) to save Petri Net models. We briefly explored the potential to extract information on processes and ICOMs from the OV-5 in a Popkin System Architect database and use this to populate a skeleton XML file for use in CPN Tools. We determined that it would be extremely difficult to fully populate the XML file as this requires translation of information for the placement of Petri Net elements on the screen. It should be relatively straightforward however to automatically perform a consistency check between a CPN XML file and a static architecture. Such a tool would be a very useful development aid when building an executable architecture.

The concept of using mechanisms in a CPN was introduced in this project. There may be other ways to implement mechanisms for analytic purposes. In our study the attributes of the organizational resources were static throughout the simulation. How would it impact the system if the organizational resources contained attributes that were dynamic during the simulation? For example, it is desired to assess the impact of fatigue of each shift working 12 hours a day. A function could be used to modify a divisions effectiveness as their shift progresses. This concept of degraded functionality could apply to any system, biological or not. Future research in this area could broaden the analytical capabilities of a CPN.

Appendix A. List of Acronyms

Table A.1 – List of Acronyms

Acronym	Description
AC	Audio Conferencing
ACF	Analysis, Correlation, and Fusion
ACO	Airspace Control Order
AE	Aeromedical Evacuation
AECT	Aeromedical Evacuation Coordination Team
AFOTTP	Air Force Operational Tactics, Techniques, and Procedures
AFWG	Architecture Framework Working Group
ALCT	Airlift Control Team
AMC	Air Mobility Command
AMCT	Air Mobility Control Team
AMD	Air Mobility Division
AOC	Air Operations Center
AOD	Air Operations Directive
ARCT	Air Refueling Control Team
ATO	Air Tasking Order
AUAB	Al-Udeid Air Base
AV	All View
C2	Command and Control
C4ISR	Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance
CAOC	Combined Aerospace Operations Center
C-C	Cognitive to Cognitive
CENTAF	Central Command Air Forces
C-I-I-C	Cognitive to Information to Information to Cognitive
CMC	Computer-Mediated Communications
COA	Course of Action
COD	Combat Operations Division
CPCL	Component Prioritized Collection List
CPD	Combat Plans Division
CPN	Colored Petri Net
CRN	Common Random Numbers
DFD	Data Flow Diagram
DoD	Department of Defense
DoDAF	Department of Defense Architecture Framework
FTF	Face-to-Face
ICOM	Input, Control, Output, Mechanism

Continued on next page

Table A.1 – continued from previous page

Acronym	Description
IDE	Intermediate Developmental Education
IDEF0	Integrated Definition for Function Modeling 0
IER	Information Exchange Requirement
I-I	Information to Information
I/O	Input/Output
IPB	Intelligence Preparation of the Battlespace
ISR	Intelligence, Surveillance, and Reconnaissance
ISRD	Intelligence, Surveillance, and Reconnaissance Division
JAOC	Joint Aerospace Operations Center
JAOP	Joint Air Operations Plan
JFACC	Joint Forces Air Component Commander
JFC	Joint Forces Commander
JIPTL	Joint Integrated Prioritized Targeting List
MAAP	Master Air Attack Plan
MOE	Measure of Effectiveness
NCW	Network Centric Warfare
OV	Operational View
PACAF	Pacific Air Forces
PAN	Process Analyzer
PED	Processing, Exploitation, and Dissemination
P-I-C	Physical to Information to Cognitive
RCC	Rescue Coordination Center
SIDO	Senior Intelligence Duty Officer
SPIN	Special Instruction
SV	Systems View
TACC	Tanker/Airlift Control Center
TBMCS	Theater Battle Management Core Systems
TET	Target Effects Team
TNL	Target Nomination List
TV	Technical View
UML	Unified Modeling Lanhuage
USAF	United States Air Force
USAFE	United States Air Forces Europe
VTC	Video Tele-conference
WS	Weapon System
XML	Extensible Markup Language

Appendix B. DoDAF Architecture Products

B.1 AV-1

- ***Architecture Project Identification***

  - *Name:* Air Tasking Order Cycle for the Air Force Aerospace Operations Center Weapon System AN/USQ-163, Falconer Architecture, Block 10.1 (as-is)
  - *Short Name:* ATO AOC WS 10.1 Architecture
  - *Organizations Developing the Architecture:* MITRE and AFIT/ENY AOC Study Thesis Group
  - *Assumptions:* Prior to Block 10, no baseline for the AOC weapon system existed; individual site organizations created and managed their own AOC's according to various local methods and designs. This architecture reflects the baseline portions of the Al-Udeid Air Base (AUAB) AOC ATO Cycle.
  - *Constraints:* Time, manpower, and no funding.
  - *Approval Authority:* Committee composed of Lt Col John M. Colombi (Chairman), Dr. David R. Jacques (Member), and Maj Joerg D. Walter (Member)
  - *Date Completed:* February 16, 2005
  - *Level of Effort and Project and Actual Costs to Develop the Architecture:*

    The development of the architecture defining the air tasking cycle was a portion of an AFIT Systems Engineering Graduate Thesis. The architecture was developed by modifying an existing architecture that MITRE produced to describe the entire AOC. This architecture was scoped for the air tasking cycle, more specifically the production and dissemination of the ATO. This was an educational endeavor that was not funded.

- ***Scope: Architecture View and Products Identification***
  - *Views and Products Available*
    - (AV-1) Overview and Summary Information
    - (AV-2) Integrated Dictionary
    - (OV-1) High Level Operational Concept
    - (OV-2) Operational Node Connectivity Description
    - (OV-3) Operational Information Exchange Matrix
    - (OV-5) Operational Activity Model
    - (OV-6a) Operational Rules Model
    - (OV-6b) Operational State Transition Description
    - (OV-7) Logical Data Model
    - (SV-1) System Interface Description
    - (SV-5) Operational Activity to Systems Function Traceability Matrix
    - (TV-1) Technical Standards Profile
  - *Time Frames Addressed:* This architecture depicts ATO cycle for the AOC WS Block 10.1 that is implemented currently through FY06.
  - *Organizations Involved*
    - AFIT/ENY
    - MITRE, ESC Divisions and Operating Locations (OL)
- ***Purpose and Viewpoint***
  - *Purpose*

    The ATO AOC WS Architecture reflects the capabilities required to produce an ATO. The Purpose of this Architecture is to support the development of an Executable Architecture for the ATO cycle in current and ”to-be” Architectures of the AOC WS. The list below details the purpose of this version.

- *From Whose Viewpoint the Architecture is Developed*

  This viewpoint belong to the AFIT System Engineering Students developing the Executable Architecture

- ***Context***

  - *Governing and Source Documents*

    * AFOTTP 2-3.2
    * AOC Familiarization Course
    * JP 3-30

- ***Tools and File Formats Used***

  MS Word, MS Excel, Popkin's System Architect were all used in developing the static architecture. File formats followed the structures laid out in DoDAF Volume II.

- ***Findings***

  - *Analysis Results* The static architecture was used to develop a CPN to evaluate the impact of different geographic configurations of the AOC. Time was selected as the measure of effectiveness assuming that different communication requirements would randomly impact the time delays of each transition. The complete methodology for this analysis can be found in Chapter Three. The model proved that having all divisions collocated is optimal from this focused perspective of analyzing communication requirements. Reference Chapter Four for the complete discussion of the results.

  - *Recommendations* Recommendations have been made in Section 5.2.

B.2 AV-2

The integrated dictionary was developed by MITRE and accompanied their architecture views. The AV-2 is available in electronic format, but due to its size it was infeasible to provide within this appendix. For more information about this view as well as other MITRE architecture products contact either Mr. Scott Surer or Mr. Ronnie Lesher.

Mr. Scott Surer
The MITRE Corporation
202 Burlington Rd. Mail Stop 1614B
Bedford, MA 01730
Phone: 781-266-9218
Email: surer@mitre.org

Mr. Ronnie Lesher
The MITRE Corporation
MITRE Gateway
Hampton, VA 23666
Phone: 757-896-8571
Email: rlesher@mitre.org

B.3 OV-1

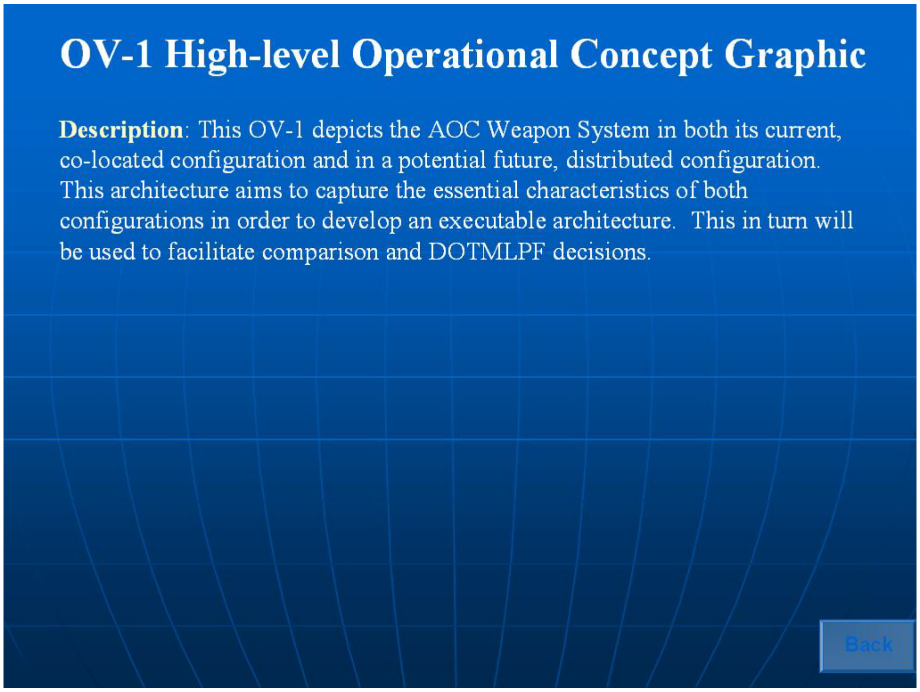


Figure B.1 OV-1 Description

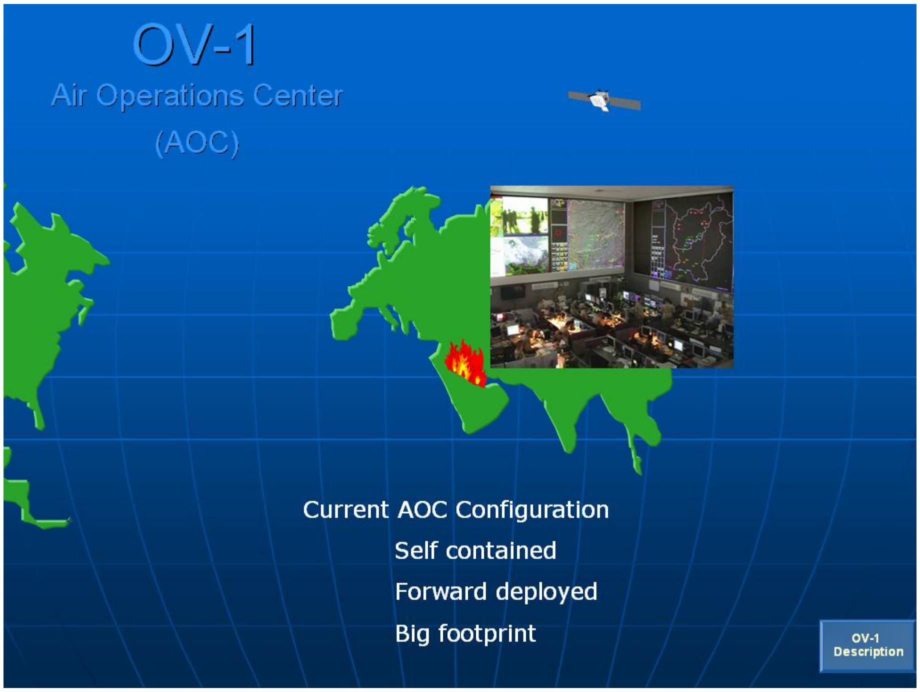


Figure B.2 OV-1 Baseline

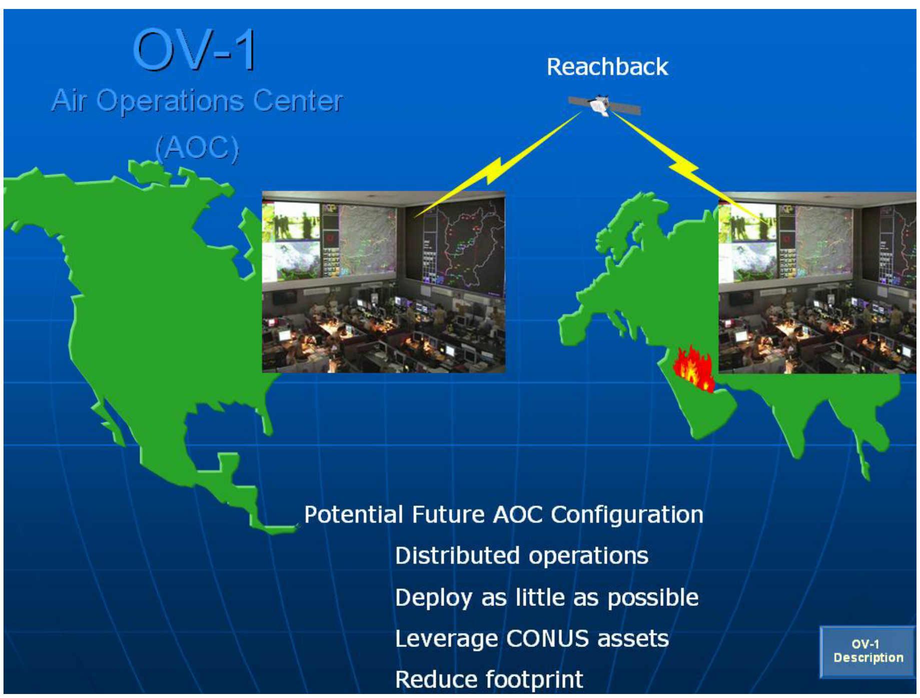


Figure B.3 OV-1 Reachback Configuration

B.4 OV-2

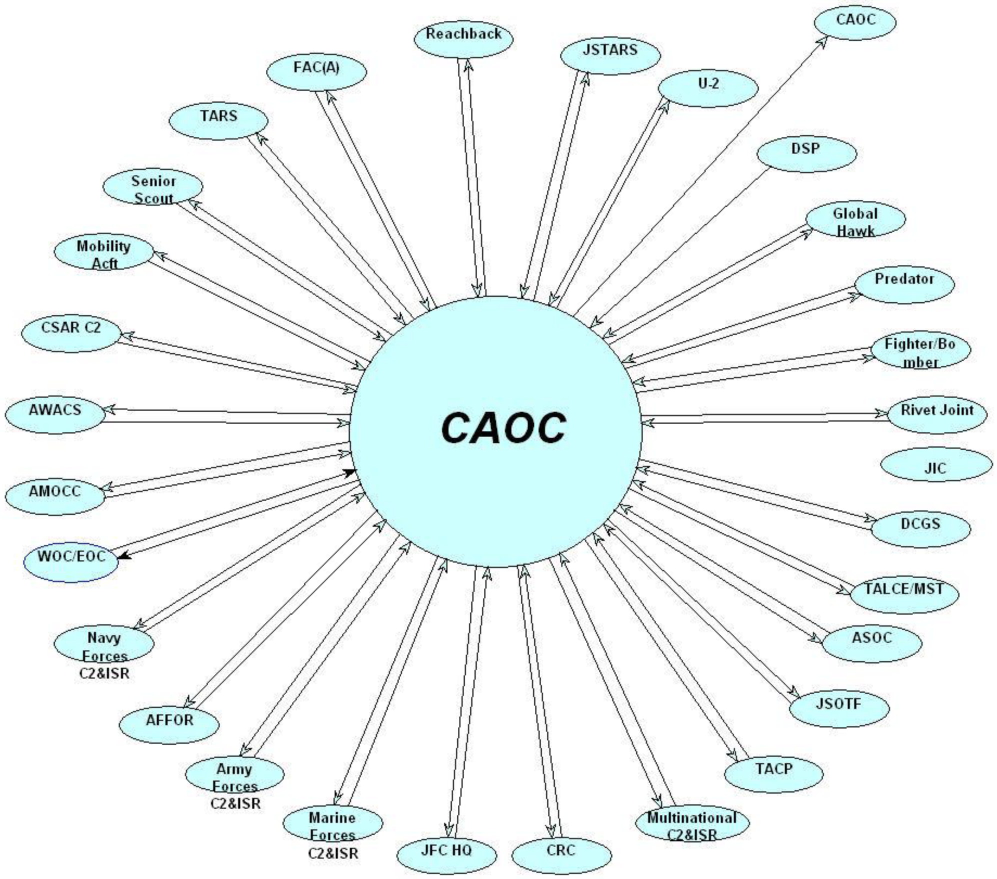


Figure B.4 OV-2

B.5 OV-3

Table B.1 OV-3

					Domain Transfers				COMM Reqts				Acceptable Mediums					
Tranisition	**Division Responsible**	**Process**	**Party 1**	**Party 2**	**C-C**	**C-I-I-C**	**I-I**	**P-I-C**	**Reliability**	**Complexity**	**Speed**	**Security**	**FACE to FACE**	**Video teleconference**	**Telephone**	**Chat Room**	**Email**	**Bulletin Board**
A11	Air Mobility	Integrates, directs execution	AMD (AMCT)	JFACC, AOC director	x				10	6	5	3	x	x	x			
A21	Air Mobility	Integrates, directs execution	AMD (AMCT)	JFACC, AOC director	x				10	6	5	3	x	x	x			
A21	Air Mobility	Coordinates aerial refueling	AMD (ARCT)	Combat Plans	x				6	4	2	3	x	x	x	x		
A321	Air Mobility	Maintains flow of assets	AMD (AECT)	Combat Plans	x				6	3	3	1	x	x	x	x	x	
A41	Air Mobility	Maintains flow of assets	AMD (AECT)	Combat Plans	x				6	3	3	1	x	x	x	x	x	
A41	Air Mobility	Coordinates aerial refueling	AMD (ARCT)	Combat Plans	x				6	4	2	3	x	x	x	x		
A41	Air Mobility	Coordinates air mobility mission into	AMD(ALCT)	Combat Plans	x				5	3	2	2	x	x	x	x	x	
A41	Air Mobility	Coordinates air mobility mission into	AMD (ARCT)	Combat Plans	x				5	3	2	2	x	x	x	x	x	
A41	Air Mobility	Coordinates air mobility mission into	AMD (AECT)	Combat Plans	x				5	3	2	2	x	x	x	x	x	
A42	Air Mobility	Maintains flow of assets	AMD (AECT)	Combat Plans	x				6	3	3	1	x	x	x	x	x	
A42	Air Mobility	Coordinates aerial refueling	AMD (ARCT)	Combat Plans	x				6	4	2	3	x	x	x	x		
A42	Air Mobility	Coordinates air mobility mission into	AMD(ALCT)	Combat Plans	x				5	3	2	2	x	x	x	x	x	
A42	Air Mobility	Coordinates air mobility mission into	AMD (ARCT)	Combat Plans	x				5	3	2	2	x	x	x	x	x	
A42	Air Mobility	Coordinates air mobility mission into	AMD (AECT)	Combat Plans	x				5	3	2	2	x	x	x	x	x	
A43	Air Mobility	Maintains flow of assets	AMD (AECT)	Combat Plans	x				6	3	3	1	x	x	x	x	x	
A43	Air Mobility	Coordinates aerial refueling	AMD (ARCT)	Combat Plans	x				6	4	2	3	x	x	x	x		
A43	Air Mobility	Coordinates air mobility mission into	AMD(ALCT)	Combat Plans	x				5	3	2	2	x	x	x	x	x	
A43	Air Mobility	Coordinates air mobility mission into	AMD (ARCT)	Combat Plans	x				5	3	2	2	x	x	x	x	x	
A43	Air Mobility	Coordinates air mobility mission into	AMD (AECT)	Combat Plans	x				5	3	2	2	x	x	x	x	x	
A44	Air Mobility	Coordinates aerial refueling	AMD (ARCT)	Combat Plans	x				6	4	2	3	x	x	x	x		
A44	Air Mobility	Coordinates air mobility mission into	AMD(ALCT)	Combat Plans	x				5	3	2	2	x	x	x	x	x	
A44	Air Mobility	Coordinates air mobility mission into	AMD (ARCT)	Combat Plans	x				5	3	2	2	x	x	x	x	x	
A44	Air Mobility	Coordinates air mobility mission into	AMD (AECT)	Combat Plans	x				5	3	2	2	x	x	x	x	x	
A45	Air Mobility	Coordinates aerial refueling	AMD (ARCT)	Coalition	x				6	7	2	3	x	x	x			
A45	Air Mobility	Coordinates air mobility mission into	AMD (ARCT)	Combat Plans	x				5	3	2	2	x	x	x	x	x	
A45	Air Mobility	Coordinates air mobility mission into	AMD (AECT)	Combat Plans	x				5	3	2	2	x	x	x	x	x	
A46	Air Mobility	Coordinates air mobility mission into	AMD (AECT)	Combat Plans	x				5	3	2	2	x	x	x	x	x	
X	Air Mobility	Identifies ISR requirements	AMD(ALCT)	ISR		x			6	3	3	6			x	x	x	
X	Air Mobility	Identifies ISR requirements	AMD (ARCT)	ISR	x				6	3	3	6	x	x	x	x	x	
A45	Air Mobility	Puts air mobility missions in AMC C2	AMD(ALCT)	TACC			x		5	2	1	2					x	x
A45	Air Mobility	Puts air mobility missions in AMC C3	AMD (ARCT)	TACC			x		5	2	1	2					x	x
A45	Air Mobility	Puts air mobility missions in AMC C4	AMD (AME)	TACC			x		5	2	1	2					x	x
A321	Combat Plans	Match tgt to platform to weapon (MAAP)	CPD (MAAP)	CPD (ATO)		x			8	4	4	6			x	x		
A322	Combat Plans	Match tgt to platform to weapon (MAAP)	CPD (MAAP)	CPD (ATO)		x			8	4	4	6			x	x		
A323	Combat Plans	Match tgt to platform to weapon (MAAP)	CPD (MAAP)	CPD (ATO)		x			8	4	4	6			x	x		
A324	Combat Plans	Match tgt to platform to weapon (MAAP)	CPD (MAAP)	CPD (ATO)		x			8	4	4	6			x	x		
A41	Combat Plans	Ensure taskings support campaign	JFACC Intent	Plans Div	x				9	8	2	7	x	x				
A46	Combat Plans	Ensure taskings support campaign	JFACC Intent	Plans Div	x				9	8	2	7	x	x				
X	ISR Division	All Source Analysis/Correlation/Fusion	ISR Chief	JFACC	x				9	10	10	10	x					
A312	ISR Division	All Source Analysis/Correlation/Fusion	ISR (ACFT)	ISR Chief		x			9	7	10	8			x			
A313	ISR Division	All Source Analysis/Correlation/Fusion	ISR (ACFT)	ISR Chief		x			9	7	10	8			x			
A314	ISR Division	All Source Analysis/Correlation/Fusion	ISR (ACFT)	ISR Chief		x			9	7	10	8			x			
A315	ISR Division	All Source Analysis/Correlation/Fusion	ISR (ACFT)	ISR Chief		x			9	7	10	8			x			
A316	ISR Division	All Source Analysis/Correlation/Fusion	ISR (ACFT)	ISR Chief		x			9	7	10	8			x			
A317	ISR Division	All Source Analysis/Correlation/Fusion	ISR (ACFT)	ISR Chief		x			9	7	10	8			x			
A41	ISR Division	All Source Analysis/Correlation/Fusion	ISR Platforms	ISR (ACFT)				x	9	3	4	6					x	
A42	ISR Division	All Source Analysis/Correlation/Fusion	ISR Platforms	ISR (ACFT)				x	9	3	4	6					x	
A43	ISR Division	All Source Analysis/Correlation/Fusion	ISR Platforms	ISR (ACFT)				x	9	3	4	6					x	
A44	ISR Division	All Source Analysis/Correlation/Fusion	ISR Platforms	ISR (ACFT)				x	9	3	4	6					x	
A45	ISR Division	All Source Analysis/Correlation/Fusion	ISR Platforms	ISR (ACFT)				x	9	3	4	6					x	
A46	ISR Division	All Source Analysis/Correlation/Fusion	ISR Platforms	ISR (ACFT)				x	9	3	4	6					x	
A47	ISR Division	All Source Analysis/Correlation/Fusion	ISR Platforms	ISR (ACFT)				x	9	3	4	6					x	
A48	ISR Division	All Source Analysis/Correlation/Fusion	ISR Platforms	ISR (ACFT)				x	9	3	4	6					x	

Table B.2 OV-3 Continued

Tranisition	Division Responsible	Process	Party 1	Party 2	Domain Transfers: C-C	C-I-C	I-I	P-I-C	COMM Reqts: Reliability	Complexity	Speed	Security	Acceptable Mediums: FACE to FACE	Video teleconference	Telephone	Chat Room	Email	Bulletin Board
A21	ISR Division	Integrates ACF/Target Development/CA/ISR Ops Internal to Combat Plans	ISR (ACFT)	Combat Plans		x			5	2	1	4			x	x	x	x
A22	ISR Division	Integrates ACF/Target Development/CA/ISR Ops Internal to Combat Plans	ISR (ACFT)	Combat Plans		x			5	2	1	4			x	x	x	x
A311	ISR Division	Integrates ACF/Target Development/CA/ISR Ops Internal to Combat Plans	ISR (ACFT)	Combat Plans		x			5	2	1	4			x	x	x	x
A312	ISR Division	Integrates ACF/Target Development/CA/ISR Ops Internal to Combat Plans	ISR (ACFT)	Combat Plans		x			5	2	1	4			x	x	x	x
A313	ISR Division	Integrates ACF/Target Development/CA/ISR Ops Internal to Combat Plans	ISR (ACFT)	Combat Plans		x			5	2	1	4			x	x	x	x
A314	ISR Division	Integrates ACF/Target Development/CA/ISR Ops Internal to Combat Plans	ISR (ACFT)	Combat Plans		x			5	2	1	4			x	x	x	x
A315	ISR Division	Integrates ACF/Target Development/CA/ISR Ops Internal to Combat Plans	ISR (ACFT)	Combat Plans		x			5	2	1	4			x	x	x	x
A316	ISR Division	Integrates ACF/Target Development/CA/ISR Ops Internal to Combat Plans	ISR (ACFT)	Combat Plans		x			5	2	1	4			x	x	x	x
A317	ISR Division	Integrates ACF/Target Development/CA/ISR Ops Internal to Combat Plans	ISR (ACFT)	Combat Plans		x			5	2	1	4			x	x	x	x
A321	ISR Division	Integrates ACF/Target Development/CA/ISR Ops Internal to Combat Plans	ISR (ACFT)	Combat Plans		x			5	2	1	4			x	x	x	x
A322	ISR Division	Integrates ACF/Target Development/CA/ISR Ops Internal to Combat Plans	ISR (ACFT)	Combat Plans		x			5	2	1	4			x	x	x	x
A323	ISR Division	Integrates ACF/Target Development/CA/ISR Ops Internal to Combat Plans	ISR (ACFT)	Combat Plans		x			5	2	1	4			x	x	x	x
A324	ISR Division	Integrates ACF/Target Development/CA/ISR Ops Internal to Combat Plans	ISR (ACFT)	Combat Plans		x			5	2	1	4			x	x	x	x
X	ISR Division	Integrates ACF/Target Development/CA/ISR Ops Internal to Combat Plans	ISR (ACFT)	Combat Ops		x			5	2	1	4			x	x	x	x
A12	ISR Division	Integrates ACF/Target Development/CA/ISR Ops Internal to Strategy	ISR (ACFT)	Strategy		x			7	2	1	4			x	x	x	x
A13	ISR Division	Integrates ACF/Target Development/CA/ISR Ops Internal to Strategy	ISR (ACFT)	Strategy		x			7	2	1	4			x	x	x	x
A21	ISR Division	Integrates ACF/Target Development/CA/ISR Ops Internal to Strategy	ISR (ACFT)	Strategy		x			7	2	1	4			x	x	x	x
A22	ISR Division	Integrates ACF/Target Development/CA/ISR Ops Internal to Strategy	ISR (ACFT)	Strategy		x			7	2	1	4			x	x	x	x
A311	ISR Division	Integrates ACF/Target Development/CA/ISR Ops Internal to Strategy	ISR (TGT/CA)	Combat Plans		x			7	7	10	8			x			
A312	ISR Division	Integrates ACF/Target Development/CA/ISR Ops Internal to Strategy	ISR (TGT/CA)	Combat Plans		x			7	7	10	8			x			
A313	ISR Division	Integrates ACF/Target Development/CA/ISR Ops Internal to Strategy	ISR (TGT/CA)	Combat Plans		x			7	7	10	8			x			
A314	ISR Division	Integrates ACF/Target Development/CA/ISR Ops Internal to Strategy	ISR (TGT/CA)	Combat Plans		x			7	7	10	8			x			

Table B.3 OV-3 Continued

					Domain Transfers				COMM Reqts				Acceptable Mediums					
Tranisition	Division Responsible	Process	Party 1	Party 2	C-C	C-I-C	I-I	P-I-C	Reliability	Complexity	Speed	Security	FACE to FACE	Video teleconference	Telephone	Chat Room	Email	Bulletin Board
A315	ISR Division	Integrates ACF/Target Development/CA/ISR Ops Internal to Strategy	ISR (TGT/CA)	Combat Plans		x			7	7	10	8			x			
A316	ISR Division	Integrates ACF/Target Development/CA/ISR Ops Internal to Strategy	ISR (TGT/CA)	Combat Plans		x			7	7	10	8			x			
A317	ISR Division	Integrates ACF/Target Development/CA/ISR Ops Internal to Strategy	ISR (TGT/CA)	Combat Plans		x			7	7	10	8			x			
A321	ISR Division	Integrates ACF/Target Development/CA/ISR Ops Internal to Strategy	ISR (TGT/CA)	Combat Plans		x			7	7	10	8			x			
A322	ISR Division	Integrates ACF/Target Development/CA/ISR Ops Internal to Strategy	ISR (TGT/CA)	Combat Plans		x			7	7	10	8			x			
A323	ISR Division	Integrates ACF/Target Development/CA/ISR Ops Internal to Strategy	ISR (TGT/CA)	Combat Plans		x			7	7	10	8			x			
A324	ISR Division	Integrates ACF/Target Development/CA/ISR Ops Internal to Strategy	ISR (TGT/CA)	Combat Plans		x			7	7	10	8			x			
A311	ISR Division	Integrates ACF/Target Development/CA/ISR Ops Internal to Strategy	ISR Ops Tear	Combat Plans	x				7	9	8	8	x	x				
A312	ISR Division	Integrates ACF/Target Development/CA/ISR Ops Internal to Strategy	ISR Ops Tear	Combat Plans	x				7	9	8	8	x	x				
A313	ISR Division	Integrates ACF/Target Development/CA/ISR Ops Internal to Strategy	ISR Ops Tear	Combat Plans	x				7	9	8	8	x	x				
A314	ISR Division	Integrates ACF/Target Development/CA/ISR Ops Internal to Strategy	ISR Ops Tear	Combat Plans	x				7	9	8	8	x	x				
A315	ISR Division	Integrates ACF/Target Development/CA/ISR Ops Internal to Strategy	ISR Ops Tear	Combat Plans	x				7	9	8	8	x	x				
A316	ISR Division	Integrates ACF/Target Development/CA/ISR Ops Internal to Strategy	ISR Ops Tear	Combat Plans	x				7	9	8	8	x	x				
A317	ISR Division	Integrates ACF/Target Development/CA/ISR Ops Internal to Strategy	ISR Ops Tear	Combat Plans	x				7	9	8	8	x	x				
A321	ISR Division	Integrates ACF/Target Development/CA/ISR Ops Internal to Strategy	ISR Ops Tear	Combat Plans	x				7	9	8	8	x	x				
A322	ISR Division	Integrates ACF/Target Development/CA/ISR Ops Internal to Strategy	ISR Ops Tear	Combat Plans	x				7	9	8	8	x	x				
A323	ISR Division	Integrates ACF/Target Development/CA/ISR Ops Internal to Strategy	ISR Ops Tear	Combat Plans	x				7	9	8	8	x	x				
A324	ISR Division	Integrates ACF/Target Development/CA/ISR Ops Internal to Strategy	ISR Ops Tear	Combat Plans	x				7	9	8	8	x	x				
A21	ISR Division	Weather	ISR (PED)	Combat Plans			x		8	3	4	6					x	
A311	ISR Division	Weather	ISR (PED)	Combat Plans			x		8	3	4	6					x	
A312	ISR Division	Weather	ISR (PED)	Combat Plans			x		8	3	4	6					x	
A313	ISR Division	Weather	ISR (PED)	Combat Plans			x		8	3	4	6					x	
A314	ISR Division	Weather	ISR (PED)	Combat Plans			x		8	3	4	6					x	
A315	ISR Division	Weather	ISR (PED)	Combat Plans			x		8	3	4	6					x	
A316	ISR Division	Weather	ISR (PED)	Combat Plans			x		8	3	4	6					x	
A317	ISR Division	Weather	ISR (PED)	Combat Plans			x		8	3	4	6					x	

Table B.4 OV-3 Continued

Tranisition	Division Responsible	Process	Party 1	Party 2	Domain Transfers				COMM Reqts				Acceptable Mediums					
					C-C	C-I-C	I-I	P-I-C	Reliability	Complexity	Speed	Security	FACE to FACE	Video teleconference	Telephone	Chat Room	Email	Bulletin Board
A11	ISR Division	Target Nomination/System Assessment/Production	ISR (TGT/CA	Strategy			x		6	2	1	4					x	x
A12	ISR Division	Target Nomination/System Assessment/Production	ISR (TGT/CA	Strategy			x		6	2	1	4					x	x
A13	ISR Division	Target Nomination/System Assessment/Production	ISR (TGT/CA	Strategy			x		6	2	1	4					x	x
A14	ISR Division	Target Nomination/System Assessment/Production	ISR (TGT/CA	Strategy			x		6	2	1	4					x	x
A15	ISR Division	Target Nomination/System Assessment/Production	ISR (TGT/CA	Strategy			x		6	2	1	4					x	x
A16	ISR Division	Target Nomination/System Assessment/Production	ISR (TGT/CA	Strategy			x		6	2	1	4					x	x
A17	ISR Division	Target Nomination/System Assessment/Production	ISR (TGT/CA	Strategy			x		6	2	1	4					x	x
A21	ISR Division	Target Nomination/System Assessment/Production	ISR (TGT/CA	Strategy			x		6	2	1	4					x	x
A22	ISR Division	Target Nomination/System Assessment/Production	ISR (TGT/CA	Strategy			x		6	2	1	4					x	x
A23	ISR Division	Target Nomination/System Assessment/Production	ISR (TGT/CA	Strategy			x		6	2	1	4					x	x
A24	ISR Division	Target Nomination/System Assessment/Production	ISR (TGT/CA	Strategy			x		6	2	1	4					x	x
A25	ISR Division	Target Nomination/System Assessment/Production	ISR (TGT/CA	Strategy			x		6	2	1	4					x	x
A21	ISR Division	Target Nomination/System Assessment/Production	ISR (TGT/CA	Combat Plans			x		6	2	1	4					x	x
A22	ISR Division	Target Nomination/System Assessment/Production	ISR (TGT/CA	Combat Plans			x		6	2	1	4					x	x
A23	ISR Division	Target Nomination/System Assessment/Production	ISR (TGT/CA	Combat Plans			x		6	2	1	4					x	x
A24	ISR Division	Target Nomination/System Assessment/Production	ISR (TGT/CA	Combat Plans			x		6	2	1	4					x	x
A25	ISR Division	Target Nomination/System Assessment/Production	ISR (TGT/CA	Combat Plans			x		6	2	1	4					x	x
A13	Strategy	Develop C/JFACC air and space estimate	STRAT DIV	ISR (A/E)		x			5	2	1	7			x			
A13	Strategy	Develop Joint Air and Space Strategy	STRAT DIV	ISR (A/E)	x				7	2	1	8	x	x	x			
A11	Strategy	Translate PRES/SECDEF, C/JFC, C/JFACC guidance …	STRAT DIV	ISR (A/E)		x			7	3	2	5			x	x	x	
A12	Strategy	Translate PRES/SECDEF, C/JFC, C/JFACC guidance …	STRAT DIV	ISR (A/E)		x			7	3	2	5			x	x	x	
A21	Strategy	Translate PRES/SECDEF, C/JFC, C/JFACC guidance …	STRAT DIV	ISR (A/E)		x			7	3	2	5			x	x	x	
A22	Strategy	Translate PRES/SECDEF, C/JFC, C/JFACC guidance …	STRAT DIV	ISR (A/E)		x			7	3	2	5			x	x	x	
A21	Strategy	Develop/coordinate daily Air Operations Directive	STRAT DIV	ISR (A/E)	x				10	8	6	7	x	x				
A22	Strategy	Develop/coordinate daily Air Operations Directive	STRAT DIV	ISR (A/E)	x				10	8	6	7	x	x				
A23	Strategy	Generate recommended apportionment decision for C/JFC	STRAT DIV	JFACC & JFC		x			9	7	6	7			x			
A312	Strategy	Determine priority, sequencing and phasing for execution of developed tasks	STRAT DIV	ISR (A/E)		x			9	7	6	7			x			
A313	Strategy	Determine priority, sequencing and phasing for execution of developed tasks	STRAT DIV	ISR (A/E)		x			9	7	6	7			x			
A314	Strategy	Determine priority, sequencing and phasing for execution of developed tasks	STRAT DIV	ISR (A/E)		x			9	7	6	7			x			
A315	Strategy	Determine priority, sequencing and phasing for execution of developed tasks	STRAT DIV	ISR (A/E)		x			9	7	6	7			x			
A316	Strategy	Determine priority, sequencing and phasing for execution of developed tasks	STRAT DIV	ISR (A/E)		x			9	7	6	7			x			

Table B.5 OV-3 Continued

					Domain Transfers				COMM Reqts				Acceptable Mediums					
Tranisition	Division Responsible	Process	Party 1	Party 2	C-C	C-I-C	I-I	P-I-C	Reliability	Complexity	Speed	Security	FACE to FACE	Video teleconference	Telephone	Chat Room	Email	Bulletin Board
A16	Strategy	Determine priority, sequencing and phasing for execution of developed tasks	STRAT DIV	ISR (A/E)		x			9	7	6	7			x			
A23	Strategy	Integrate functional/service component task requirements into ATO	STRAT DIV	ISR (A/E)		x			9	7	6	7			x			
A13	Strategy	Develop alternative contingency plans and COAs	STRAT DIV	JFACC & JFC	x				6	4	4	7	x	x	x			
A14	Strategy	Develop alternative contingency plans and COAs	STRAT DIV	JFACC & JFC	x				6	4	4	7	x	x	x			
A15	Strategy	Develop alternative contingency plans and COAs	STRAT DIV	JFACC & JFC	x				6	4	4	7	x	x	x			
A16	Strategy	Develop alternative contingency plans and COAs	STRAT DIV	JFACC & JFC	x				6	4	4	7	x	x	x			

B.6 OV-5

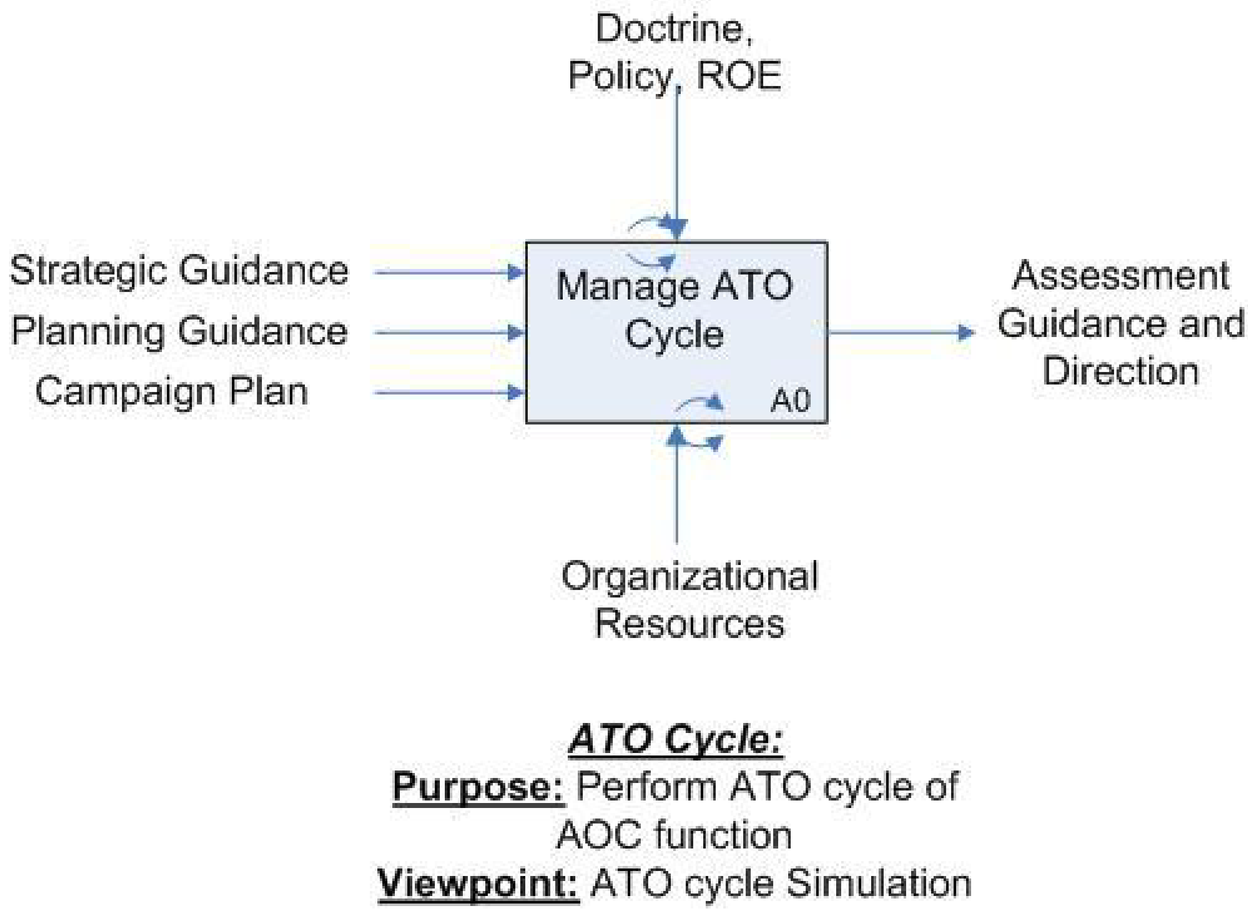


Figure B.5 OV-5 A-0

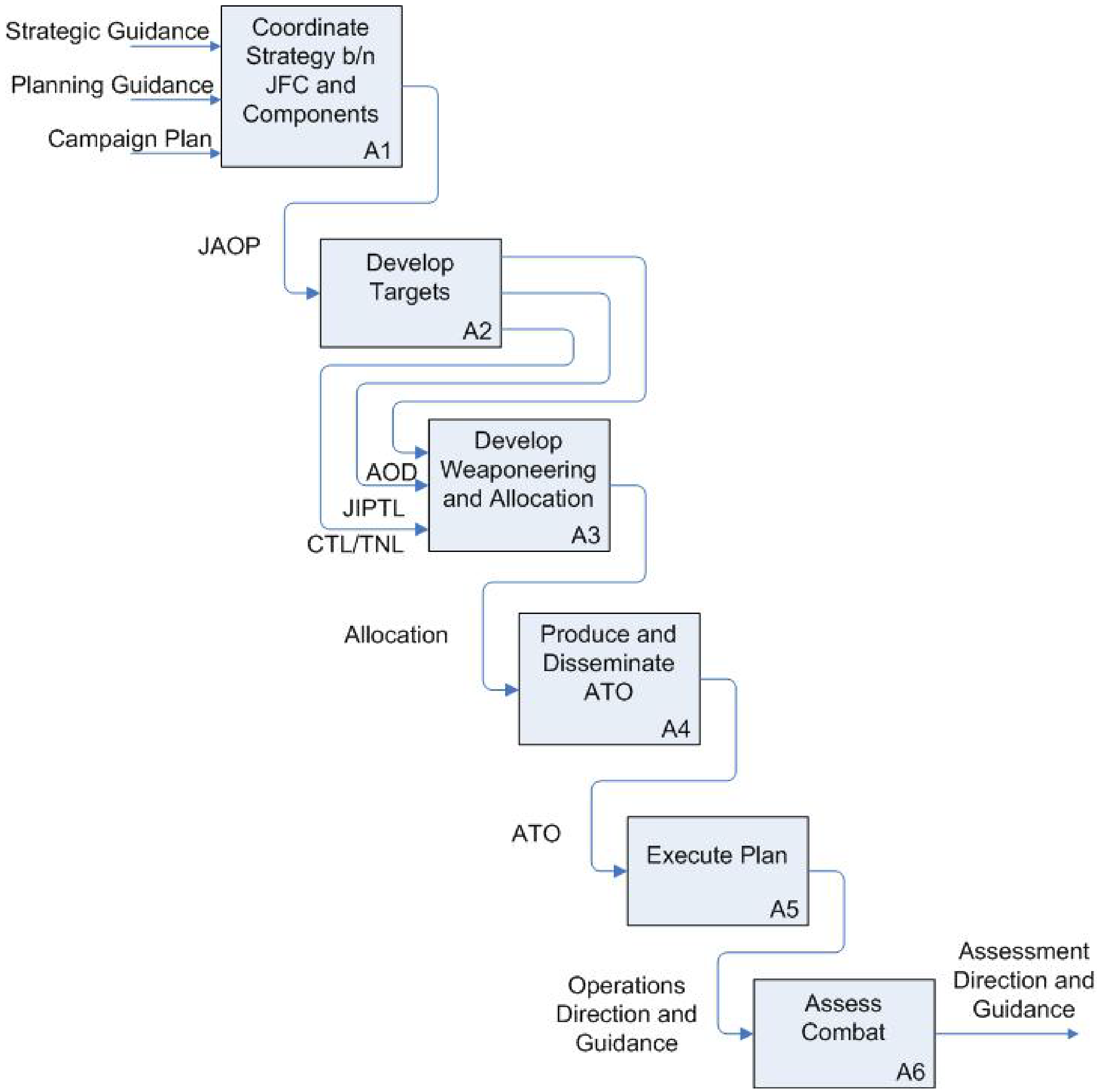


Figure B.6 OV-5 A0

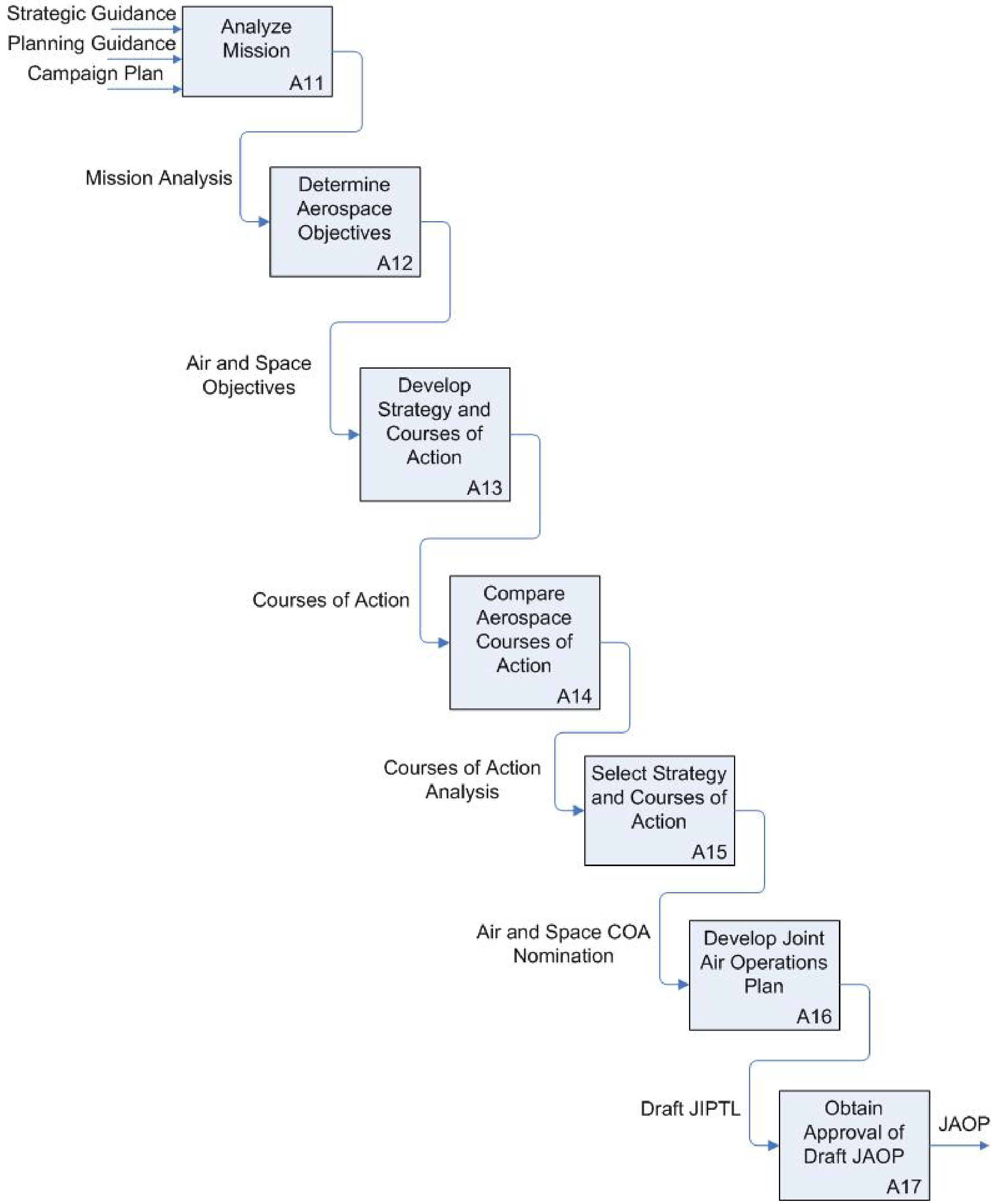


Figure B.7 OV-5 A1

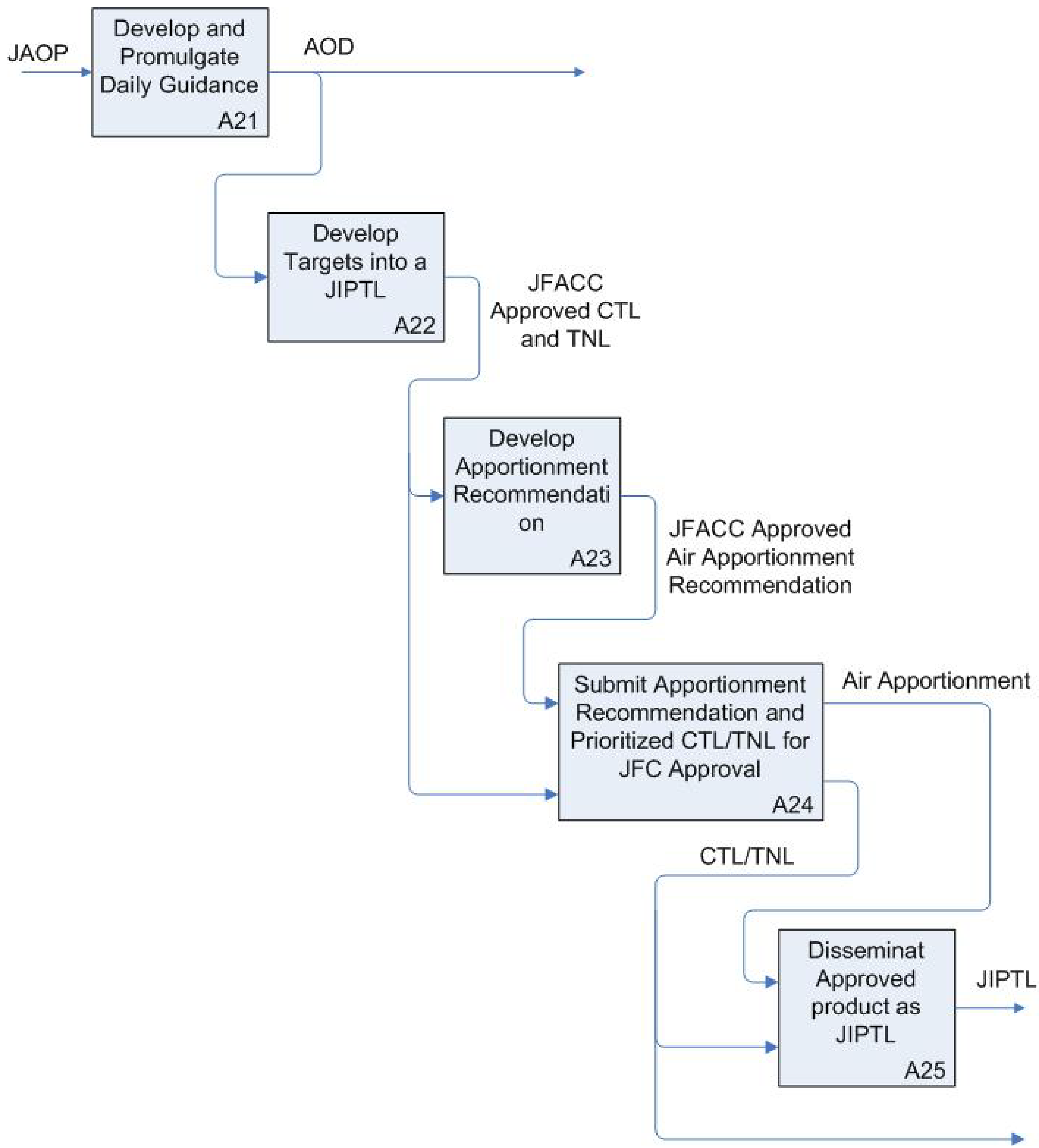


Figure B.8 OV-5 A2

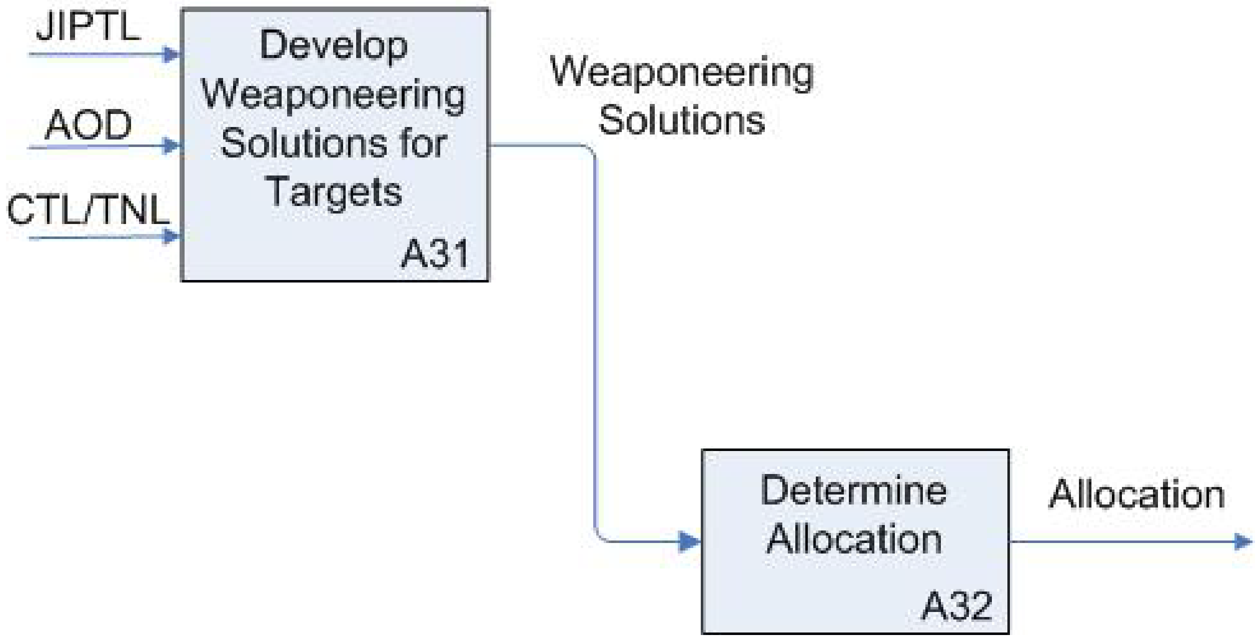


Figure B.9 OV-5 A3

B.7 OV-6a

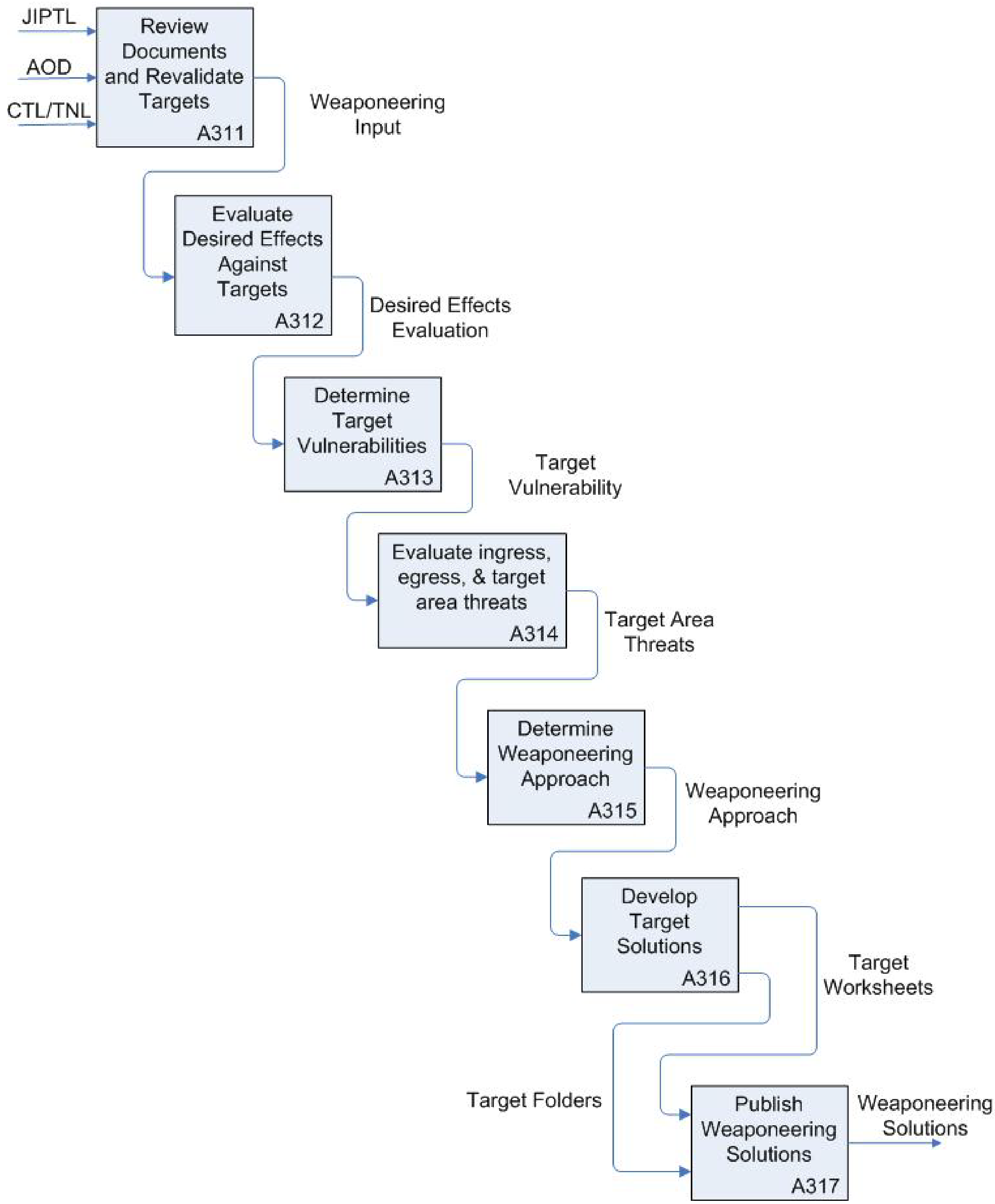


Figure B.10 OV-5 A31

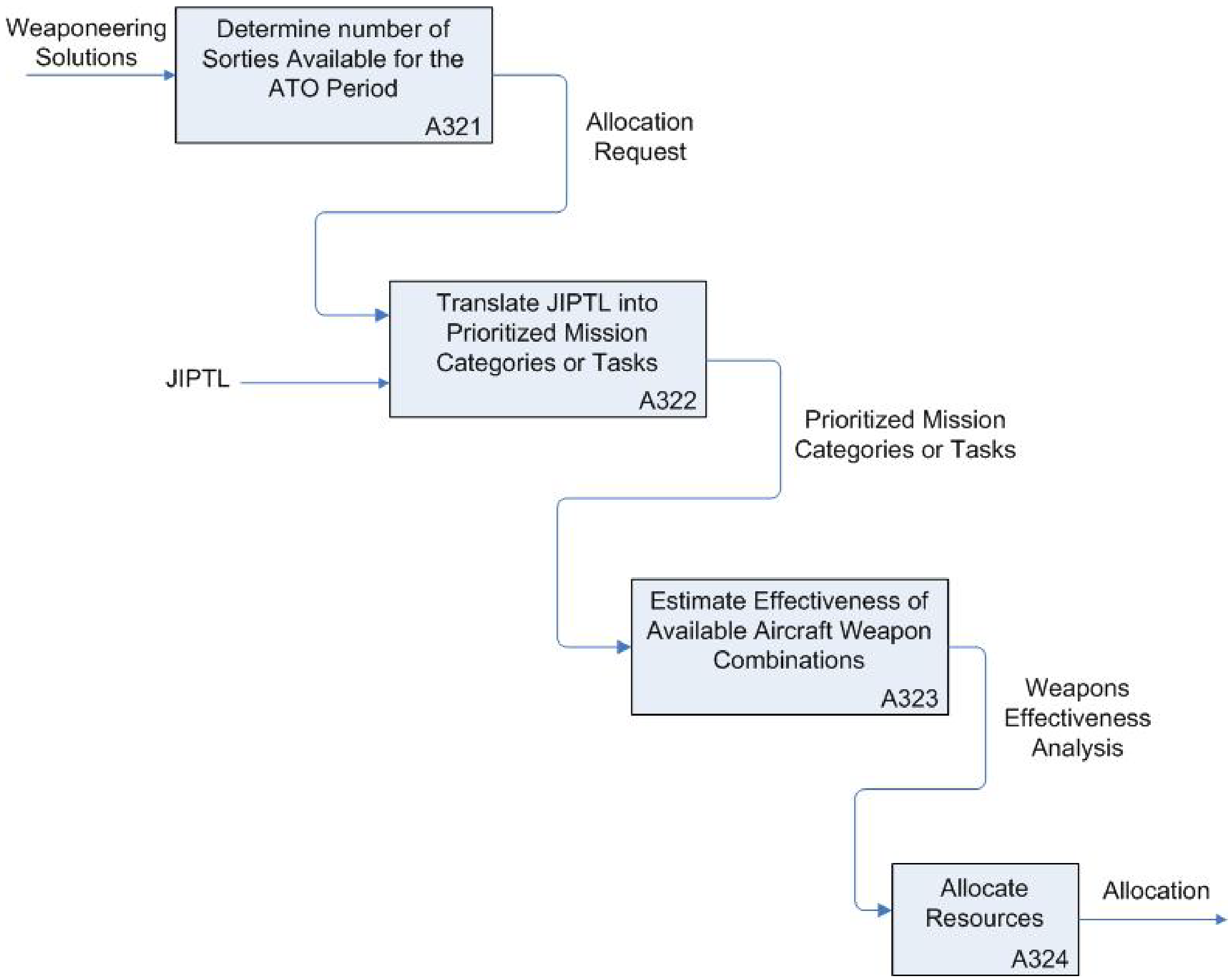


Figure B.11 OV-5 A32

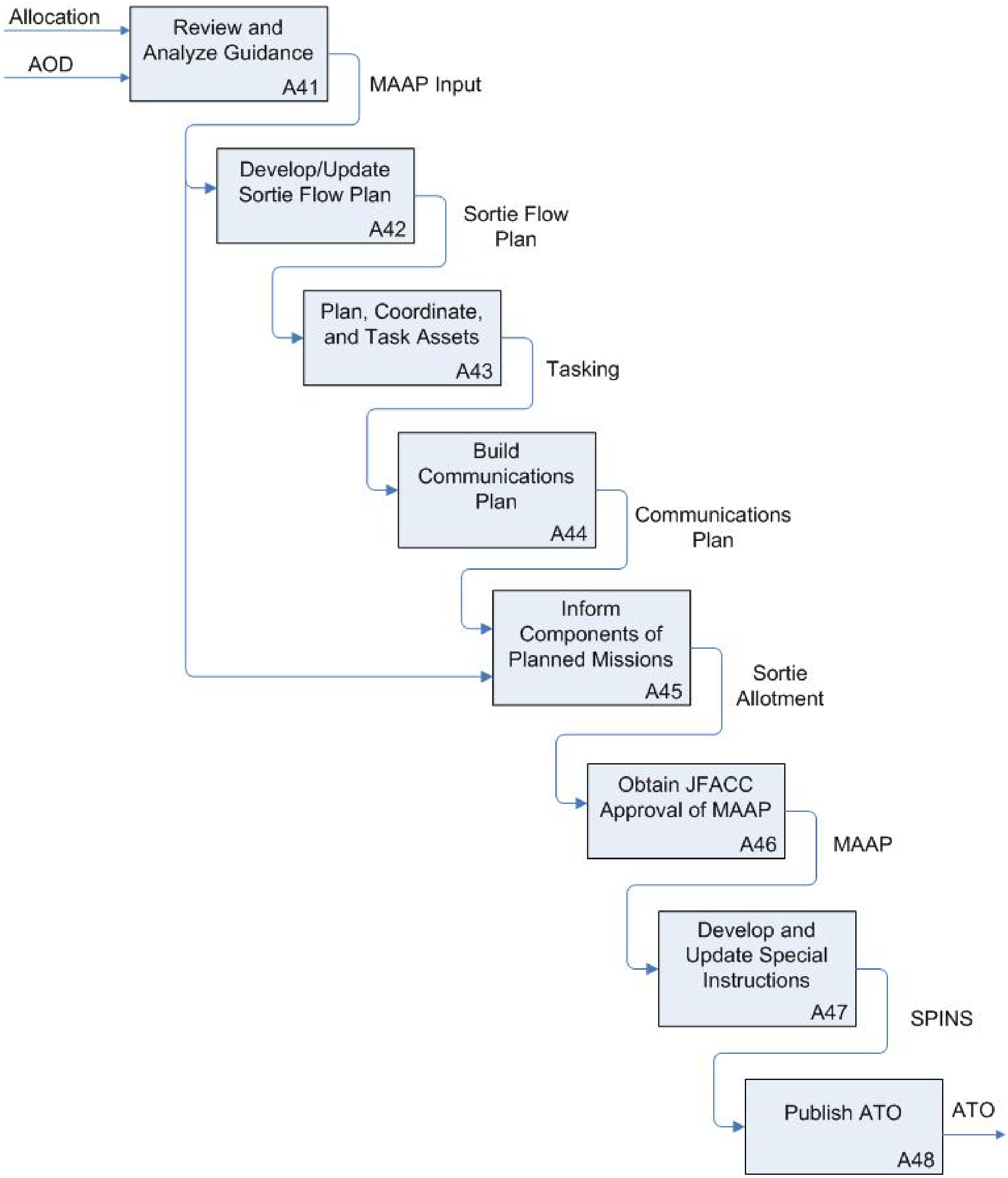


Figure B.12 OV-5 A4

Table B.6 OV-6a Process Rules

Rule Number	Structured English Explaination	Applicable Operational Activity Number	Applicable Entities
1	IF Strategic Guidance, Planning Guidance, and Campaign Plan are available THEN produce Mission Analysis	A11- Analyze Mission	Strategic Guidance, Planning Guidance, Campaign Plan, and Mission Analysis
2	IF Mission Analysis is available THEN produce Air and Space Objectives and Associated Tasks	A12- Determine Aerospace Objectives	Mission Analysis, and Air and Space Objectives and Associated Tasks
3	IF Air and Space Objectives and Associated Tasks are available THEN produce Courses of Action	A13- Develop Strategy/Course of Action	Air and Space Objectives and Associated Tasks and Courses of Action
4	IF Courses of Action are available THEN produce Courses of Action Analysis	A14- Compare Aerospace Courses of Action	Courses of Action and Courses of Action Analysis
5	IF Courses of Action Analysis is available THEN produce Air and Space COA Nomination	A15- Select Strategy/COA	Courses of Action Analysis and Air and Space COA Nomination
6	IF Air and Space COA Nomination is available THEN produce JAOP Draft	A16- Develop JAOP	Air and Space COA Nomination and JAOP Draft
7	IF JAOP Draft is available THEN produce JAOP	A17- Obtain Approval of Detailed JAOP	JAOP Draft and JAOP
8	IF JAOP is available THEN produce the AOD	A21- Develop and Promulgate Daily Guidance	JAOP and AOD
9	IF AOD is available for the current ATO cycle THEN produce JFACC Approved CTL/TNL	A22- Develop Targets into a JIPTL	AOD and JFACC Approved CTL/TNL
10	IF JFACC Approved CTL/TNL is available THEN produce JFACC Approved Air Apportionment Recommendation	A23- Develop Apportionment Recommendation	JFACC Approved CTL/TNL and JFACC Approved Air Apportionment Recommendation
11	IF JFACC Approved Air Apportionment Recommendation AND JFACC Approved CTL/TNL are available THEN produce Air Apportionment AND CTL/TNL	A24- Submit Apportionment Recommendation and Prioritized CTL/TNL for JFC Approval	JFACC Approved Air Apportionment Recommendation, JFACC Approved CTL/TNL, Air Apportionment, and CTL/TNL
12	IF Air Apportionment AND CTL/TNL are available THEN produce the JIPTL	A25- Disseminate Approved Product as JIPTL	Air Apportionment, CTL/TNL, and JIPTL

Table B.7 OV-6a Process Rules Continued

Rule Number	Structured English Explaination	Applicable Operational Activity Number	Applicable Entities
13	IF AOD, CTL/TNL, and JIPTL are available THEN produce Weaponeering Input	A311- Review Documents and Revalidate Targets	AOD,CTL/TNL, JIPTL, and Weaponeering Input
14	IF Weaponeering Input is available THEN produce Desired Effects Evaluation	A312- Evaluate Desired Effects Against Targets	Weaponeering Input and Desired Effects Evaluation
15	IF Desired Effects Evaluation is available THEN produce Target Vulnerabilities	A313- Determine Target Vulnerabilities	Desired Effects Evaluation and Target Vulnerabilities
16	IF Target Vulnerabilities is available THEN produce Target Area Threats	A314- Evaluate Ingress and Egress and Target Area Threats	Target Vulnerabilites and Target Area Threats
17	IF Target Area Threats is available THEN produce Weaponeering Approach	A315- Determine Weaponeering Approach	Target Vulnerabilities and Target Area Threats
18	IF Weaponeering Approach is available THEN produce Target Worksheets AND Target Folders	A316- Develop Target Solutions	Weaponeering Approach, Target Worksheets, and Target Folders
19	IF Target Worksheets AND Target Folders are available THEN produce Weaponeering Solution	A317- Publish Weaponeering Solutions	Target Worksheets, Target Folders, and Weaponeering Solution
20	IF Weaponeering Solutions are available THEN produce ALLOREQ	A321- Determine Number of Sorties Available	Weaponeering Solutions and ALLOREQ
21	IF the JIPTL AND ALLOREQ are available THEN produce Prioritized Mission Categories/Tasks	A322- Translate JIPTL into Prioritized Mission Categories or Tasks	JIPTL, ALLOREQ, and Prioritized Mission Categories/Tasks
22	IF Prioritized Mission Categories/Tasks is available THEN produce Weapons Effectiveness Analysis	A323- Estimate Effectiveness of Available Aircraft-Weapon Combinations	Prioritized Mission Categories/Tasks, and Weapons Effectiveness Analysis
23	IF Weapons Effectiveness Analysis is available THEN produce Allocation	A324- Allocate Resources	Weapons Effectiveness Analysis and Allocation
24	IF Allocation and AOD are available THEN produce MAAP Input	A41- Review and Analyze Guidance	Allocation, AOD, and MAAP Input
25	IF MAAP Input is available THEN produce Sortie Flow Plan	A42- Develop Sortie Flow Plan	MAAP Input and Sortie Flow Plan
26	IF Sortie Flow Plan is available THEN produce Tasking	A43- Plan, Coordinate and Task Assets	Sortie Flow Plan and Tasking

Table B.8 OV-6a Process Rules Continued

Rule Number	Structured English Explaination	Applicable Operational Activity Number	Applicable Entities
27	IF Tasking is available THEN produce Communications Plan	A44- Build Communications Plan	Tasking and Communications Plan
28	IF Communications Plan AND MAAP Input are available THEN produce Sortie Allotments	A45- Inform Components of Planned Missions	Communications Plan, MAAP Input, and Sortie Allotments
29	IF Sorite Allotments are available THEN produce MAAP	A46- Obtain FJACC Approval of the MAAP	Sortie Allotments and MAAP
30	IF MAAP is available THEN produce SPINs	A47- Develop/Update Special Instructions	MAAP and SPINs
31	IF SPINs are available THEN produce ATO	A48- Publish ATO	SPINs and ATO
32	IF ATO is available THEN produce Operations Direction and Guidance	A5- Execute Plan	ATO and Operations Direction and Guidance
33	IF Operations Direction and Guidance is available THEN produce Assessment Guidance and Direction	A6- Assess Combat	Operations Direction and Guidance and Assessment Guidance and Direction

Table B.9 Time Delay Rules

Configuration	SD Location	CPD Location	AMD Location	Delay
1	Deployed	Deployed	Deployed	Missed Req'ts * discrete(1,5)
2	Deployed	Deployed	TACC	Missed Req'ts * discrete(1,15)
3	Reachback	Reachback	Deployed	Missed Req'ts * discrete(1,15)
4	Reachback	Reachback	TACC	Missed Req'ts * discrete(1,30)

B.8 OV-6b

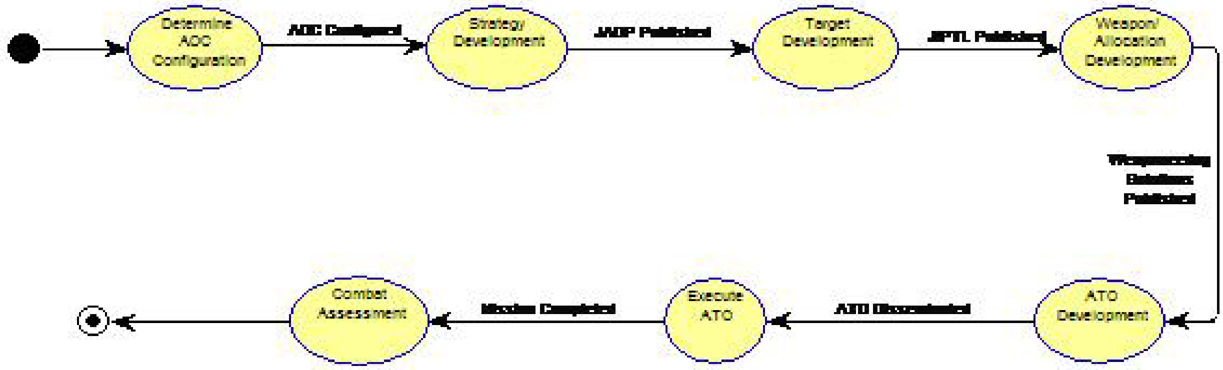


Figure B.13 OV-6b Air Tasking Cycle State Transition Diagram

B.9 OV-7

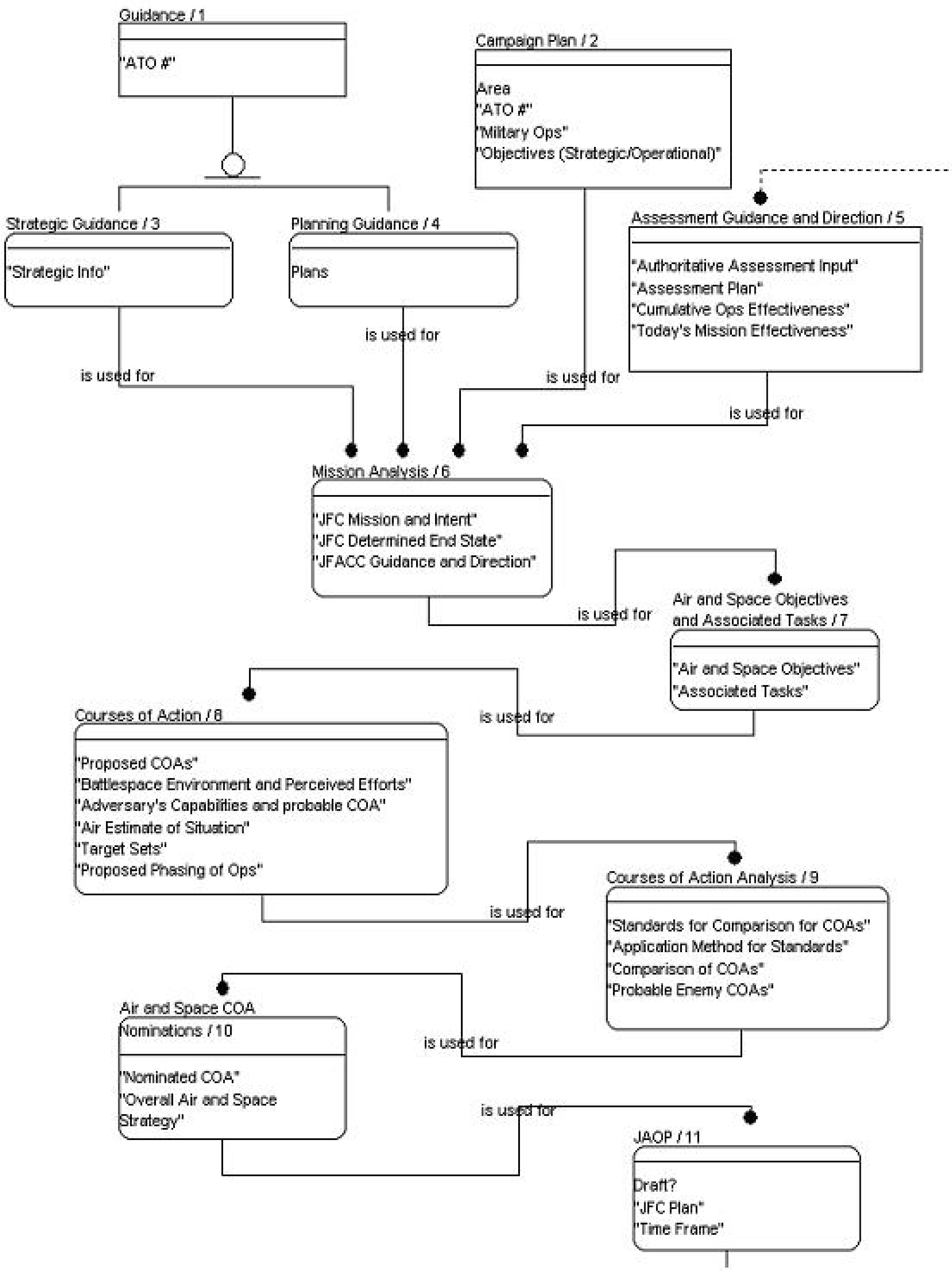


Figure B.14 OV-7 Entities Corresponding to A1

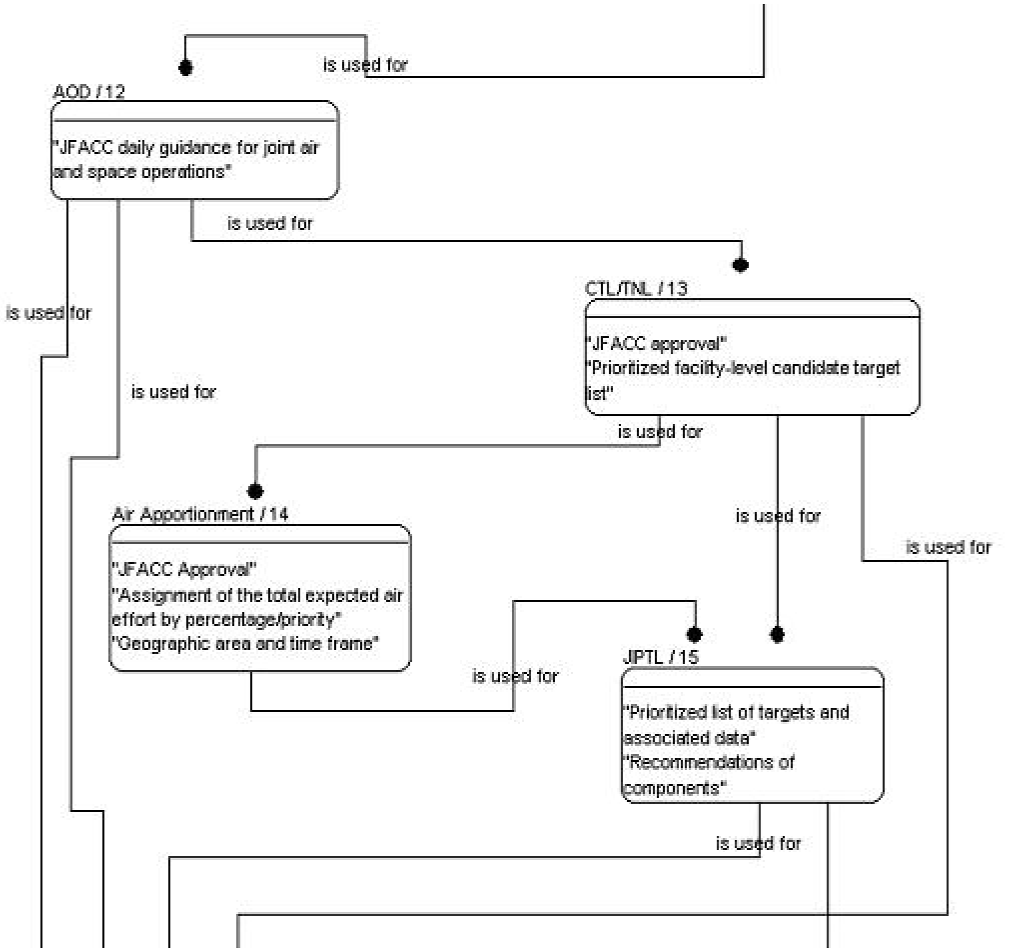


Figure B.15 OV-7 Entities Corresponding to A2

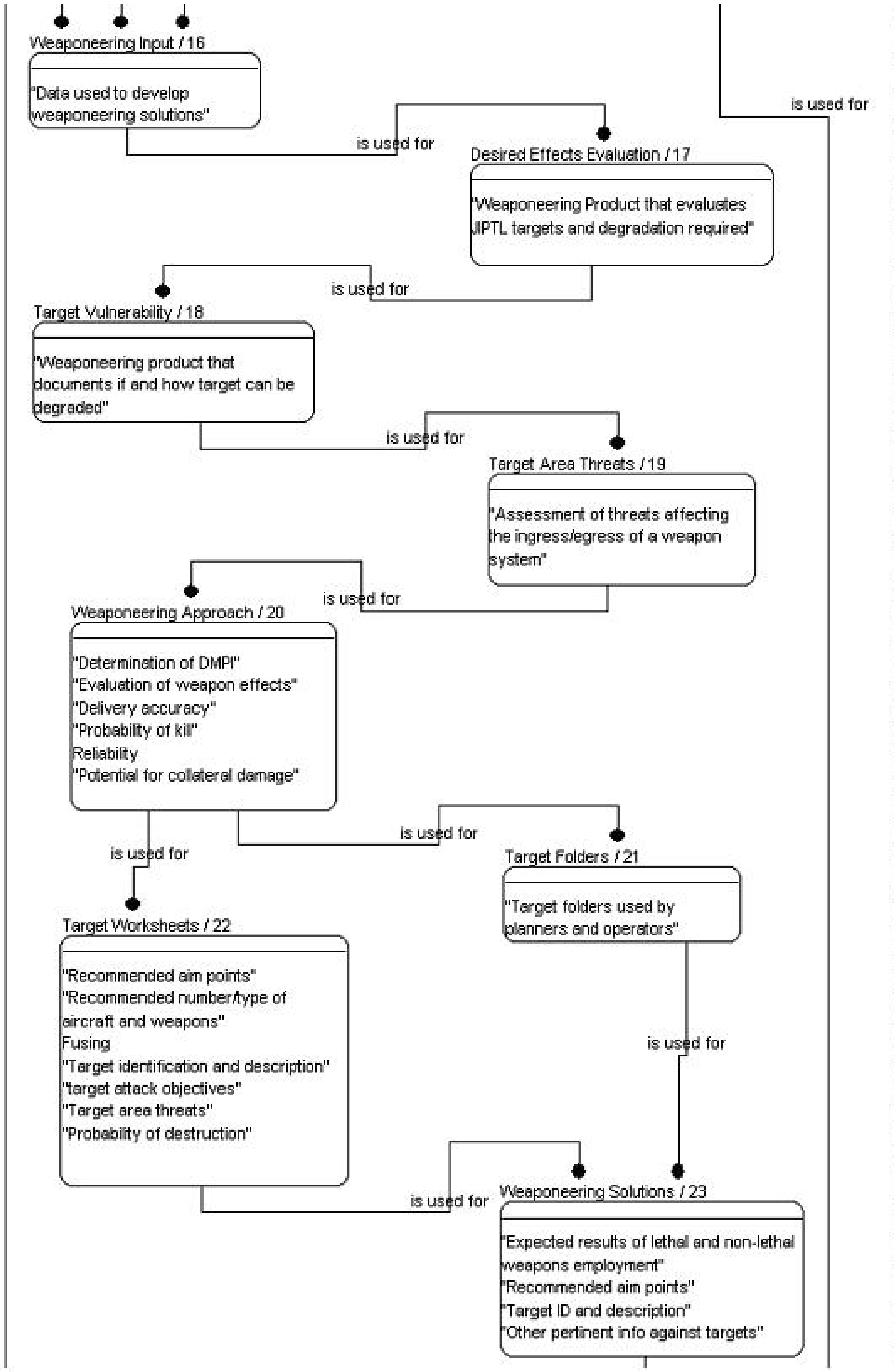


Figure B.16 OV-7 Entities Corresponding to A31

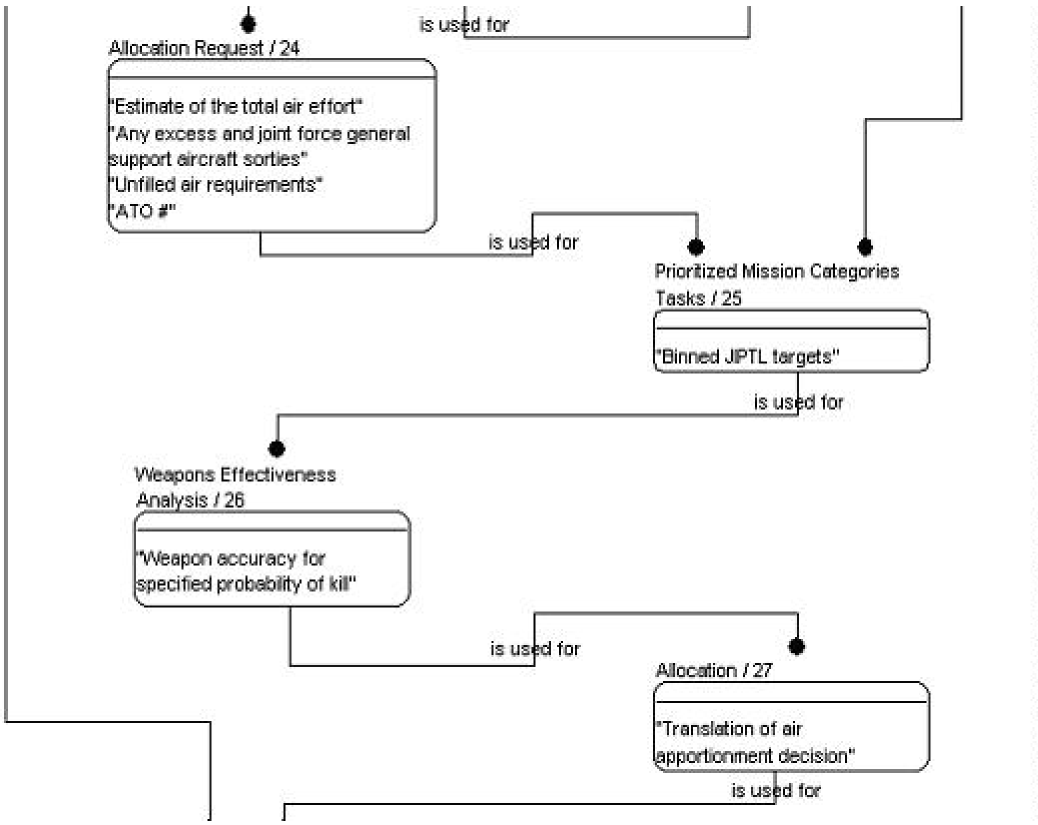


Figure B.17 OV-7 Entities Corresponding to A32

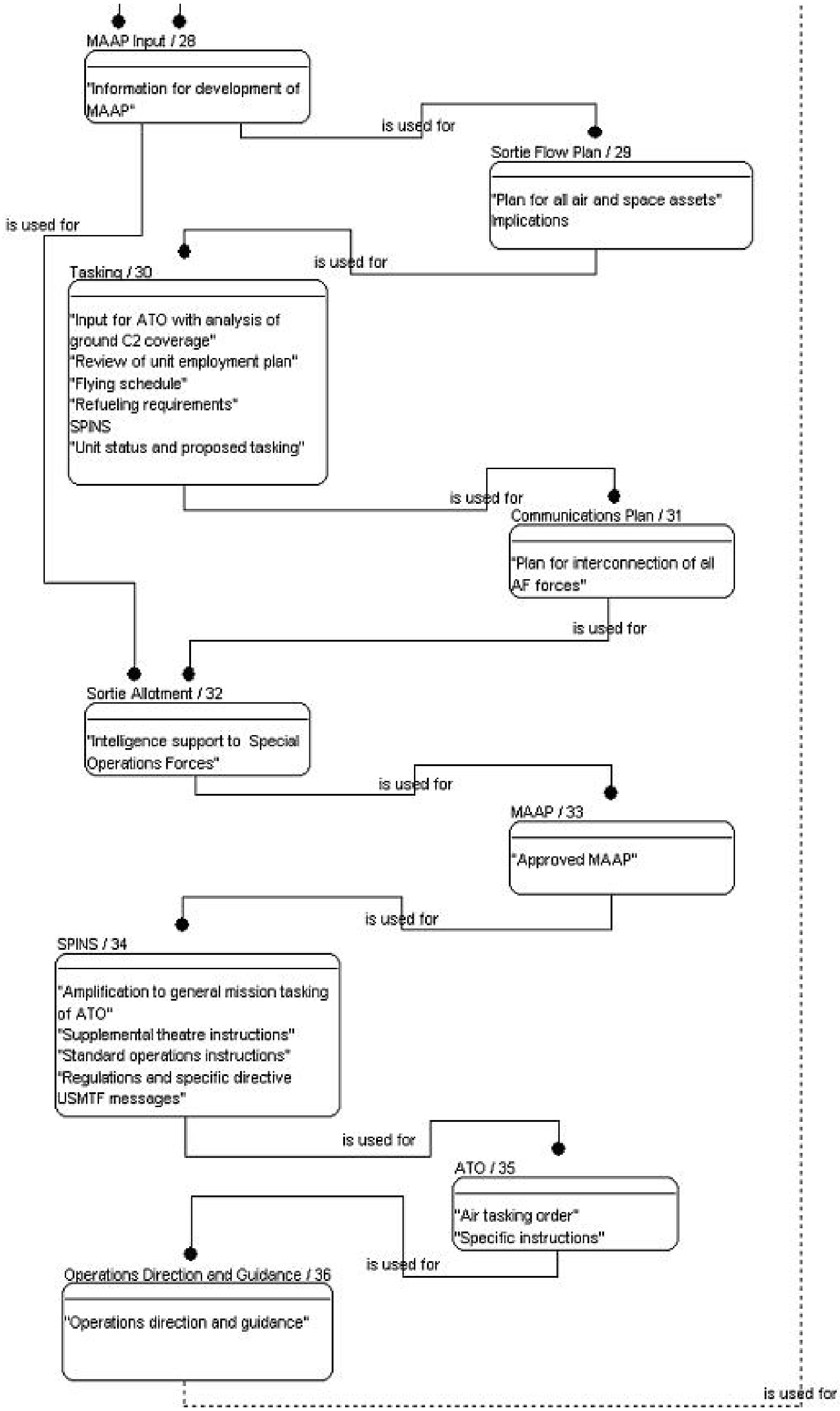


Figure B.18 OV-7 Entities Corresponding to A4

B.10 SV-1

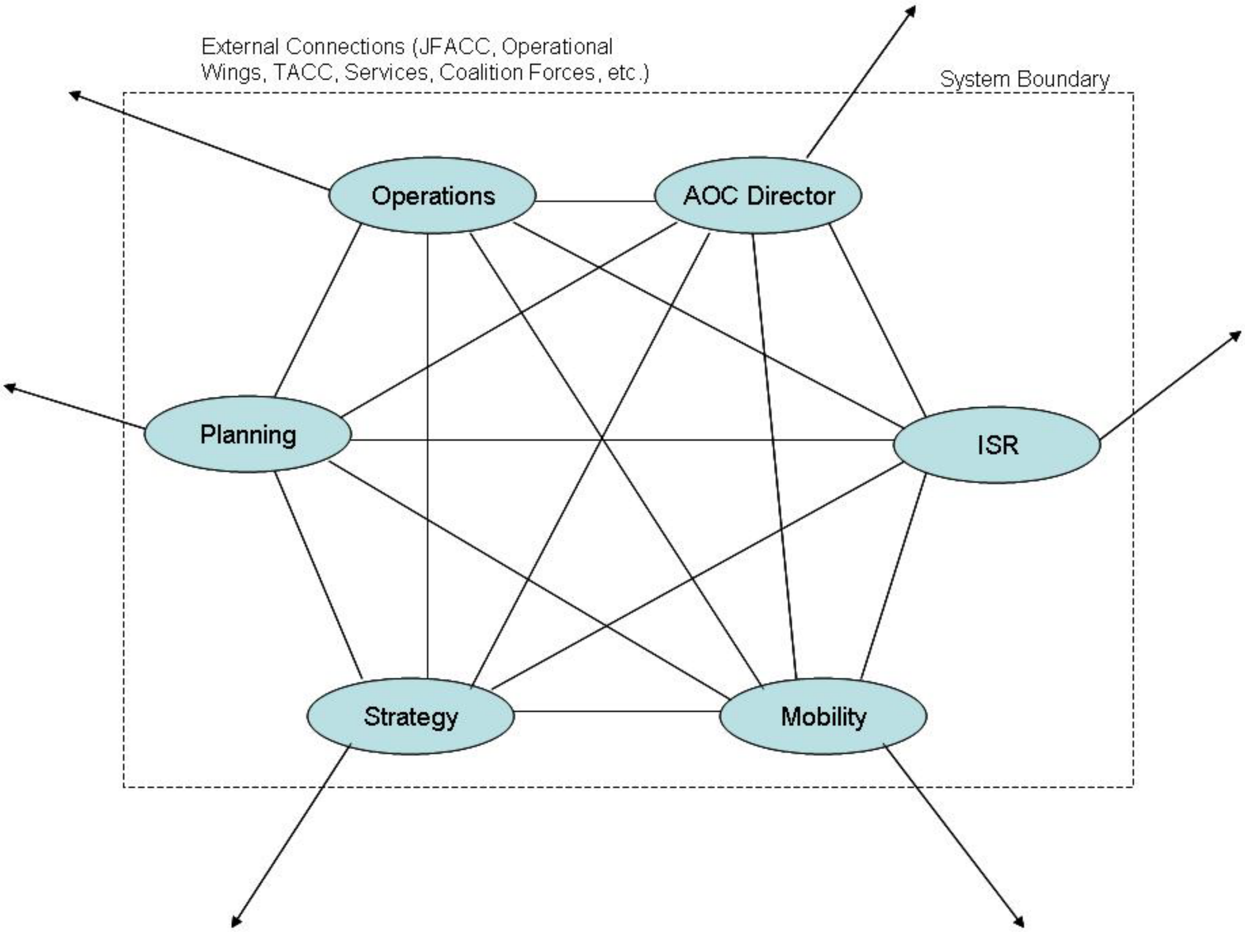


Figure B.19 SV-1 Baseline

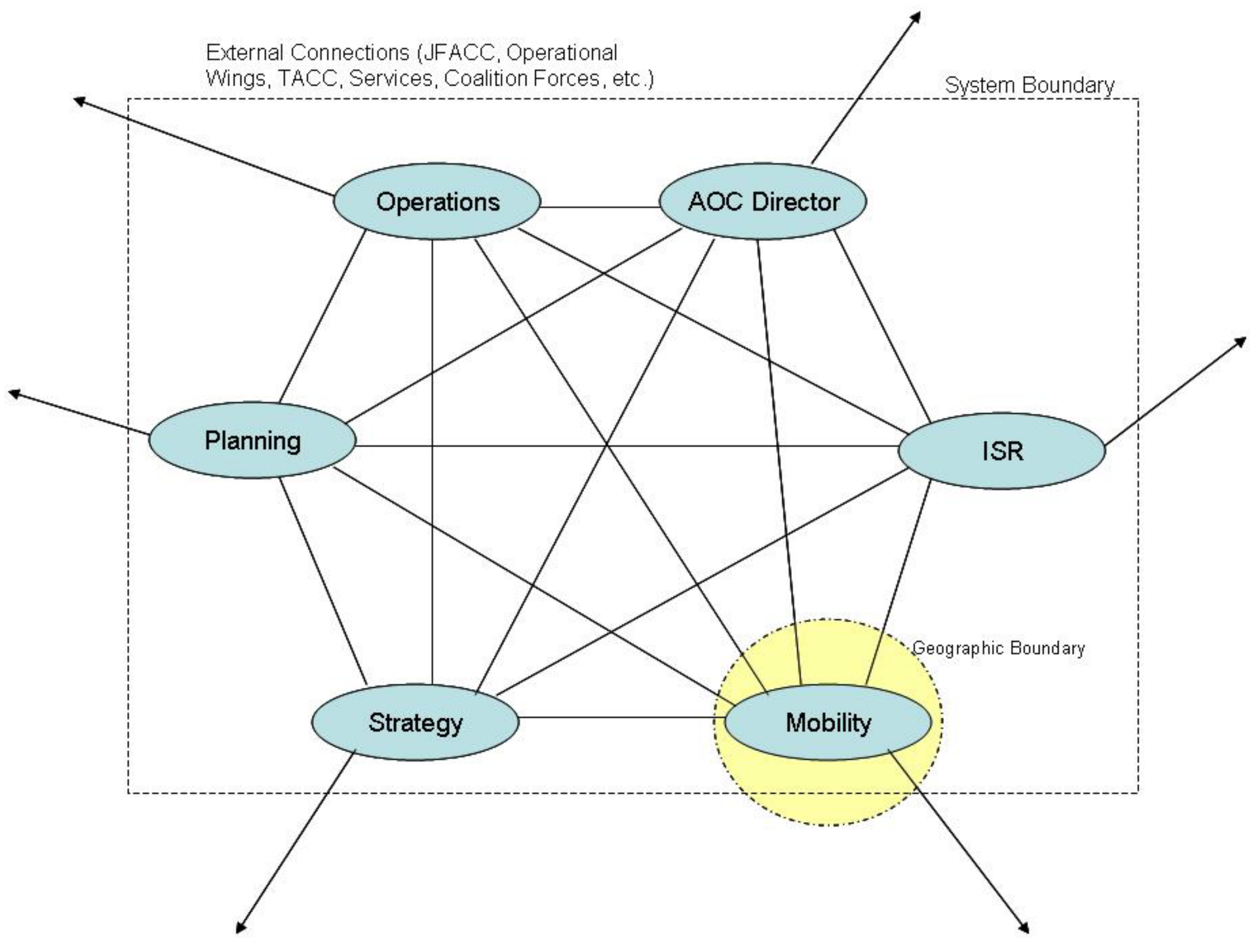


Figure B.20 SV-1 Configuration Two

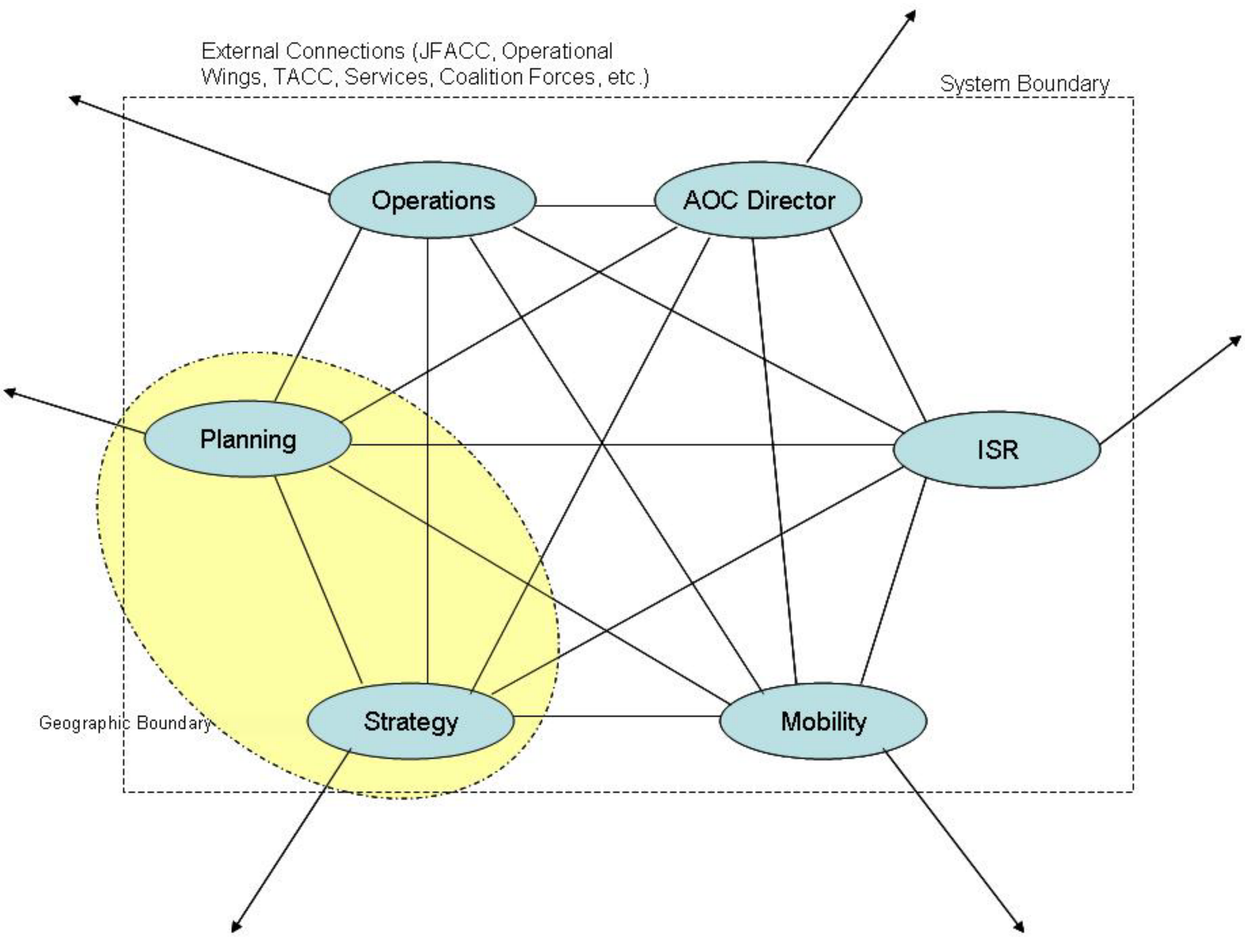


Figure B.21 SV-1 Configuration Three

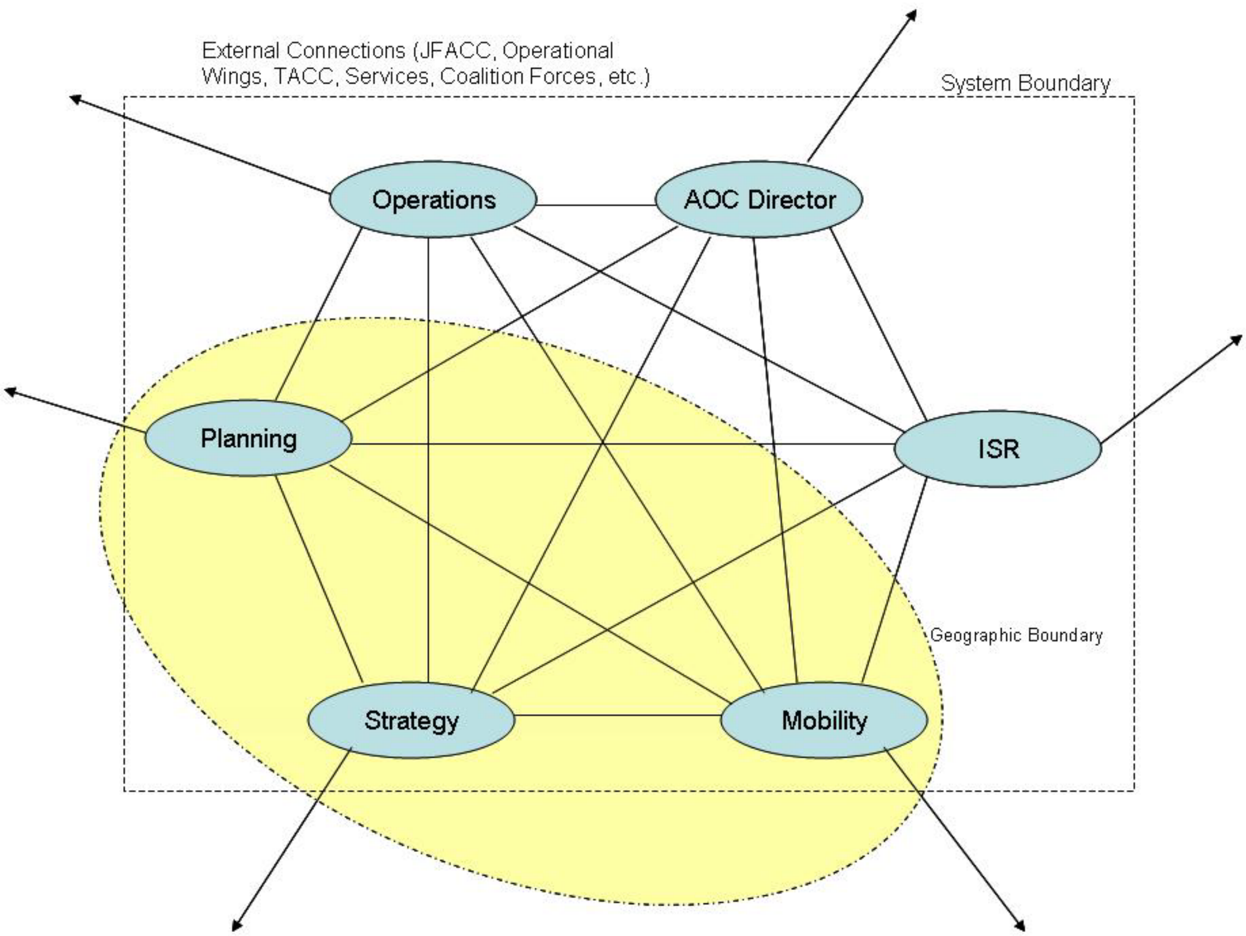


Figure B.22 SV-1 Configuration Four

B.11 SV-5

SV-5 Relationships

Operational Activity	System Function	System
A1.1-Analyze Mission	Create Mission Analysis	MS Office/DMS/TBMCS/IRIS/GCCS/S-VTC
A1.2-Determine Aerospace Objectives	Prioritize objectives and assign tasks	MS Office/DMS/TBMCS/JDP/JTT/ITS/IWPC/CST/JCE-IWS
A1.3-Develop Strategy/Courses of Action	Prepare COA	IWS/GTN/JDISSGDSS/Broadsword/Tel-scope/IW Vis/CMMA
A1.5-Select Strategy/COA	Edit briefings	Microsoft Powerpoint
A1.6-Develop JAOP	Edit briefings	Microsoft Powerpoint/Word/S-VTC
A1.7-Obtain Approval of Detailed JAOP	Secure VTC	S-VTC
A2.1-Develop and Promulgate Daily Guidance	Create AOD	MS Office/JDISS/CMMA/ITS/JTT/TBMCS/SBMCS/JDP/AD/S-VTC
A2.2-Develop Targets into a JIPTL	Create CTL/TNL	MS Office/ITS/JTT
A2.4-Submit Apportionment Recommendation and Prioritized CTL/TNL for JFC Approval	Disseminate JIPTL	ITS
A2.5-Disseminate Approved Product as JIPTL	Transfer target lists across security domains	TBMCS/ISDS/ITS
A3.1.1-Review Documents and Revalidate Targets	Edit briefings/documents	Microsoft PP/Word
A3.1.2-Evaluate desired effects against targets	Evaluate kinetic/non-kinetic options	ITS/JTT
A3.1.3-Determine Target Vulnerabilities	Evaluate Options	ITS/JTT/IWPC/Tel-scope
A3.1.4-Evaluate ingress, egress and target area threats	Update ATF	ITS
A3.1.5-Determine weaponeering approach	Evaluate Options	ITS/JTT
A3.1.6.1-Develop Target Solutions	Access ATF/Mensurate targets/Select coordinates	ITS/JTT/RainDrop
A3.1.7-Publish weaponeering solutions	Transfer target lists across security domains	ITS/JTT
A3.2.1-Determine Number of Sorties Available	Process Air Support Requests	TBMCS/WARP
A-3.2.2-Translate JIPTL into Prioritized Mission Categories or Tasks	Manage JIPTL data	ITS/JTT
A-3.2.3-Estimate Effectiveness of Available Aircraft-Weapon Combinations	Threat model analysis	MS Office/JTT/SBMCS
A-3.2.4-Allocate Resources	Develop ADP/Edit briefings, spreadsheet	TBMCS/JDP/PP/Excel
A-4.1-Review and Analyze Guidance	Edit documents	MS Office/ITS/JTT/TBMCS/SBMCS/AD/JDISS
A-4.2-Develop/Update Sortie Flow Plan	Edit spreadsheets/Share files	Microsoft Excel
A-4.3-Plan, Coordinate and Task Assets	Develop ADP/ACP/ATO/SPINS	TBMCS/JDP/AD/TAP/ITS/JTT/WARP/JDISS/CMMA
A-4.4-Build Communications Plan	Develop ATO/SPINS	TBMCS/TAP
A-4.5-Obtain JFACC approval of MAAP	Edit briefings/Secure VTC	Microsoft Powerpoint/S-VTC
A-4.6-Obtain JFACC approval of MAAP	Secure VTC	S-VTC
A-4.6-Develop/Update Special Instructions	Develop SPINS	TBMCS/TAP
A-5.0-Execute Plan	Perform Battlefield Operations	Many
A-6.0-Assess Combat	Data access	ITS

Figure B.23 SV-5

B.12 TV-1

This is an excerpt from the MITRE technical standards. The complete list of technical standards is well over 70 pages and does not impact the thrust of the research in this thesis. As such, the complete list of technical standards will not be shown.

Table B.10 TV-1 Excerpt

Information Security

Service	Standards
JTA-6.2 Mandated Standards	ISO-IEC 15408:1999; Information Technology Security Techniques Evaluation Criter
	ISO-IEC 15408:1999; Information Technology Security Techniques Evaluation Criter
JTA-6.2.2.1 Application Software Entity Security Standards	FORTEZZA Application Implementers Guide; MD40021011.52; 5 March 1996.
	FORTEZZA Cryptologic Interface Programmers Guide (CIPG); Revision 1.52; 30 Janua
	FORTEZZA Application Implementers Guide; MD40021011.52; 5 March 1996.
	FORTEZZA Cryptologic Interface Programmers Guide (CIPG); Revision 1.52; 30 Janua
JTA-6.2.2.2.1.Authentication Security Standards	IETF RFC 1510; The Kerberos Network Authentication Service; Version 5; 10 Septem
	FIPS PUB 112; Password Usage; 30 May 1985.
	IETF RFC 1510; The Kerberos Network Authentication Service; Version 5; 10 Septem
	FIPS PUB 112; Password Usage; 30 May 1985.

Bibliography

1. "Air Force Operational Tactics, Techniques, and Procedures 2-3.2- Air and Space Operations Center."
2. *Arena Professional Edition Reference Guide*. Systems Modeling Corporation, 2000.
3. "IEEE 1471-2000 Recommended Practice for Architectural Description of Software-Intensive Systems," October 2000.
4. "DoD Instruction 5000.2 Mandatory Procedures for Major Defense Acquisition Programs," April 2002.
5. "Chairman of the Joint Chiefs of Staff Instruction 3170.01C," June 2003.
6. "Department of Defense Architectural Framework Volume I," August 2003.
7. "Department of Defense Architectural Framework Volume II," August 2003.
8. A. Pinsonneault, K. L. Kraemer. "The Impact of Technological Support on Groups: An Assessment of the Empirical Research," *Decision Support Systems*, *5*:197–216 (1989).
9. Alberts, David S. *Information Age Transformation*. Command and Control Research Program, 2003.
10. Alexander Levis, Lee Wagenhals. "C4ISR Architectures: I. Developing a Process for C4ISR Architecture Design," *Systems Engineering*, *3*(4) (2000).
11. Averill Law, W. David Kelton. *Simulation Modeling and Analysis*. McGraw-Hill, 1999.
12. Buede, Dennis. *The Engineering Design of Systems*. John Wiley & Sons, 2000.
13. David S. Alberts, John J. Garstka, Fredrick P. Stein. *Network Centric Warfare-Developing and Leveraging Information Superiority*. DoD Command and Control Research Program, 2003.
14. Durrett, Richard. *Probability: Theory and Examples*. Duxbury Press, 1995.
15. E. Isaacs, J. Tang. "What Video Can and Cant Do for Collaboration: A Case Study," *Multimedia Systems*, *2*:63–73 (1994).
16. et. al., Baude. *Network Centric Analysis of Air Operations*. MS thesis, Air Force Institute of Technology, 2004.
17. H. H. Clark, S. E. Brennan. *Perspectives on Socially Shared Cognition*. American Psychological Association, 1991.
18. L. Wainfan, P. Davis. *Challenges in Virtual Collaboration*. Rand Corporation, 2004.

19. Lee Wagenhals, Insub Shin, Daesik Kim Alexander Levis. "C4ISR Architectures: II. A Structured Analysis Approach for Architecture Design," *Systems Engineering*, *3*(4) (2000).

20. Norman, Douglas. "Engineering a Complex System: A Study of the AOC," (2004).

21. Rechtin, E. *Systems Architecting: Creating & Building Complex Systems*. Prentice Hall, 1991.

22. Rechtin, E. "The Art of Systems Architecting," *IEEE Spectrum*, 66–69 (October 1992).

23. Reid, A. A. L. *The Social Impact of the Telephone*. MIT Press, 1977.

24. S. G. Straus, J. A. Miles, L. L. Levesque. "The Effects of Videoconference, Telephone, and Face-to-Face Media on Interviewer and Applicant Judgments in Employment Interviews," *Journal of Management*, *27*(3):363–381 (2001).

25. Steven Dam, James Willis. *Defensible C4ISR Architectures Require the Application of Good System Engineering Practice*. Systems and Proposal Engineering Company, 2002.

26. Stimpson, D., "Minimizing forward footprint interview," 2004.

27. W. David Kelton, Randall P. Sadowski, Deborah A. Sadowski. *Simulation With Arena* (2 Edition). McGraw-Hill, 2002.

28. Wackerly, Dennis. *Mathematical Statistics with Applications*. Duxbury Press, 2001.

29. Zinn, Andrew. *The Use of Integrated Architectures to Support Agent Based Simulation: An Initial Investigation*. MS thesis, Air Force Institute of TEchnology, 2004.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 074-0188*

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
Grad date 21 Mar 2005	**Master's Thesis**	6 Oct 2003 – 21 Mar 2005

4. TITLE AND SUBTITLE

EXECUTABLE ARCHITECTURES AND THEIR APPLICATION TO A GEOGRAPHICALLY DISTRIBUTED AIR OPERATIONS CENTER

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Robert J. Beal, Captain, USAF
Jeremy P. Hendrix, Captain, USAF
Garth P. McMurray, Captain, USAF
William C. Stewart, Civilian, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GSE/ENY/05-M03

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFC2ISRC/DOX
Attn: Mr. David Hess
Langley AFB, Va 23665 DSN: 575-4155

10. SPONSOR/MONITOR'S ACRONYM(S)

N/A

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

N/A

12. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Integrated Architectures and Network Centric Warfare represent two central concepts in DoD's on-going transformation. The true power of integrated architectures is brought to bear when they are combined with simulation to move beyond a static representation and create an executable architecture. This architecture can then be used to experiment with system configurations and parameter values to guide employment decisions. This thesis applies the methodology of Dr. Alexander Levis, former Chief Scientist of the Air Force, to the static architecture representing the Aerospace Operations Center (AOC). Using Colored Petri Nets and other simulation tools, an executable architecture for the AOC's Air Tasking Order (ATO) production thread was developed. These models were then used to compare the performance of a current, forward deployed AOC configuration to three other potential configurations, which utilize a network centric environment to deploy a portion of the AOC and provide reach-back capabilities to the non-deployed units. Performance was measured by the amount of time required to execute the ATO cycle under each configuration. Communication requirements were analyzed for each configuration and stochastic delays were modeled for all transactions in which requirements could not be met due to the physical configuration of the AOC elements. All four configurations were found to exhibit statistically different behavior with regard to ATO cycle time.

15. SUBJECT TERMS

Executable Architectures, Colored Petri nets, Communication Theory, Air Operations Center, Air Tasking Cycle, Air Tasking Order

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					John Colombi, Lt Col, USAF
REPORT	**ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(Include area code)*
U	**U**	**U**	**UU**	178	(937) 255-3355 ext. 3347; e-mail: john.colombi@afit.edu